arXiv:2304.13896v1 [cs.LO] 27 Apr 2023

结构感知下界与拓宽量化布尔公式可处理性视野

Johannes K. Fichte*、Robert Ganian†、Markus Hecher‡、Friedrich Slivovsky†和Sebastian Ordyniak§
*计算机与信息科学系（IDA），林雪平大学，SE-581 83，林雪平，瑞典
†维也纳工业大学逻辑与计算研究所，Favoritenstraße 9–11，1040维也纳，奥地利
‡麻省理工学院，美国马萨诸塞州剑桥市，MA 02139§利兹大学计算学院，
英国利兹，LS2 9JT

摘要——QSAT 问题要求对量化布尔公式（QBF）求值，在近似、计数、判定和概率复杂性研究中具有基础性意义，也被视为典型的 PSPACE 完全问题。因此，此前研究者已在各种结构限制（参数）下对其进行研究，其中最受关注的是基于实例原始图表示的参数化。事实上，已知即使将实例限制为原始图树宽为常数，QSAT 仍然是 PSPACE 完全的；但以顶点覆盖数（原始图）为参数时，该问题存在一个双指数固定参数算法。

然而，既有研究尚未充分揭示：从其他自然的实例表示方式，尤其是通过关联图的视角来看，QSAT 的复杂性究竟如何。本文提出了结构感知归约，使我们能够针对高度受限的 QSAT 实例获得几乎紧的下界，其中包括关联图具有有界树深度或反馈顶点数的实例。我们进一步提出了 QSAT 的新算法，以补充这些下界；这些算法在标准图论参数化下，几乎完整地刻画了该问题的复杂性。我们还展示了对其他自然图表示的影响，并针对原始图的更细粒度参数化，获得了 QSAT 的新上界和下界。

I. 引言

量化布尔公式的求值问题（QSAT）是布尔可满足性问题（SAT）的自然推广，也是理论计算机科学中最重要的问题之一，应用于符号推理 [6], [8], [22], [45]，[50], [56], 约束满足问题（CSP） [30]，[17], [28], 数据库和逻辑 [33]。QSAT 的输入公式由一个（量词）前缀和一个矩阵组成，其中矩阵可以是任意布尔公式，但通常假设其为合取范式（CNF），例如通过经典的 Tseytin 变换转换而成。QSAT 被视为 PSPACE 完全问题的典型代表，并已从经典近似 [63],计数 [61], 判定 [47], 和概率复杂性 [40], 的角度得到广泛研究，同时也从参数化复杂性的视角进行了研究 [46], [3]。

针对 QSAT 研究的绝大多数参数化都依赖于矩阵的一种合适图表示；事实上，这与布尔可满足性问题（SAT） [53], [55], [64], 整数线性规划（ILP） [29], [11], 约束满足问题（CSP） [28], [18], [54], 及其他基础问题的情况相似。对于 QSAT，文献中考虑得最经典的参数化，是公式合取范式（CNF）矩阵的原始图表示的树宽 k。如今，这方面的复杂度已得到充分理解：即使仅以该参数进行参数化，问题仍然是 PSPACE 完全的 [46], [3], [26]，即便将范围限制在路径分解上也是如此；不过，当 ℓ 表示公式前缀的量词深度、n 表示公式中的变量数时，可以在 $\mathrm{tow}(\ell, k)\cdot\mathrm{poly}(n)$[1] 时间内求解。更积极的一点是，已知仅以原始图的顶点覆盖数进行参数化，可以为 QSAT [38] 带来一个固定参数算法，其运行时间为双指数级。

树宽与顶点覆盖数在参数依赖性上存在 ℓ 重指数级差距，这引出了一个问题：当不再将量词深度 ℓ 作为参数时，固定参数可处理性的边界在哪里？在参数化复杂性中，存在一整套比树宽限制更强、比顶点覆盖数限制更弱的结构参数层级，其中最突出的是树深度 [43] 以及反馈顶点数和反馈边数[2]。然而，还有一个更大的空白：我们所知甚少

作者按字母顺序排列。Fichte 和 Hecher 在访问 Simons Institute 期间开展了这项工作。本研究得到奥地利科学基金（FWF）项目 J4656、P32830 和 Y1329、下奥地利研究资助协会（GFF）项目 ExzF-0004、维也纳科学技术基金（WWTF）项目 ICT19-060 和 ICT19-065，以及由瑞典政府资助的 ELLIIT 的支持。

[1]运行时间关于树宽 k呈指数增长，其中 k 位于高度为量词深度 ℓ 的迭代指数塔顶端。[2]分别表示到无环性的顶点删除距离或边删除距离。

关于 QSAT 在矩阵表示（原始图除外）背景下的复杂性理论图景。此类图表示中最突出的例子是关联图，它已在 SAT [53], [57], [55],、CSP [54], [15], 和 ILP [23], 等问题中得到广泛研究。上述 QSAT 困难性也从原始图树宽延续到了关联图的树宽 [3]；该问题在以量词深度加关联图树宽为参数时是固定参数可处理的 [10], , 但此前尚无关于该图结构参数的其他结果。

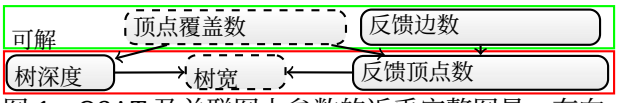


图 1：QSAT 及关联图上参数的近乎完整图景；有向弧表示源参数是目标参数的上界，参见例如 [58], [16]。红框表示具有不可解性的情形，其下界与已知上界基本匹配（对于树宽和反馈顶点数而言与指数时间假设紧密匹配）；绿框表示可处理性（固定参数可处理）结果。粗体文字标示新结果。

A. 贡献概述

受第三节中用于获得关于树宽 [26], 的 QSAT 下界的高层次方法启发，我们形式化定义了 QSAT 的结构感知（SAW）归约概念。这些归约可用于精确展示输入实例参数与归约后实例参数之间的函数依赖关系。利用结构感知归约的概念，我们在第四节中为具有高度限制的 QSAT 实例类别建立了紧下界；这些结果对于矩阵的三种不同表示具有深远的复杂性理论含义。这些结果实质上排除了针对树深度的高效算法，以及使用反馈顶点数时快于树宽算法的算法。下面我们重点介绍这些结果，随后分别讨论每种表示形式。除非指数时间假设（ETH） [34] 成立：

1）除非指数时间假设（ETH）不成立，否则对于量词深度 ℓ为线性的情况，QSAT 无法在低于 $\mathrm{tow}(\ell', k) \cdot \mathrm{poly}(n)$ 的时间内求解，其中 k 或者是关联图的反馈顶点数，或者是关联图的树深度。2）除非指数时间假设（ETH）不成立，否则对于量词深度 ℓ为线性的情况，QSAT 无法在低于 $\mathrm{tow}(\ell', k) \cdot \mathrm{poly}(n)$ 的时间内求解，其中 k 或者是合取范式（CNF）与析取范式（DNF）组合公式的原始图的反馈顶点数，或者是该图的树深度。3）除非指数时间假设（ETH）不成立，否则对于量词深度 ℓ 为线性的情况，QSAT 无法在低于 $\mathrm{tow}(\ell', k) \cdot \mathrm{poly}(n)$ 的时间内求解，其中 k 或者是从矩阵中删除单个子句后的原始图的反馈顶点数，或者是该图的树深度。

1. 关联图上的结果。我们的两个复杂性理论下界表明，就关联图这一基本表示而言，QSAT 的不可解性边界明显低于树宽。这也引出了一个问题：我们能否利用对关联图施加更强限制的参数，为该问题获得高效算法？迄今为止最自然的两个结构图满足这些性质的参数是前文提到的顶点覆盖数和反馈边数。

我们还为 QSAT 提供了关于这两个参数的固定参数算法作为下界的补充，详见第五节。由此，我们基于关联图的结构，几乎完整地描绘了该问题的复杂性，见图 1。

2. 组合矩阵的原始图。以往对 QSAT 的复杂性理论研究通常假设矩阵以 CNF 形式表示，这种形式支持标准的图表示，并且可以通过经典的 Tseytin 变换 [62]从任意公式得到。然而，实证证据表明，范式转换会对 QSAT 求解器的性能产生不利影响 [1], , 而如今的求解器通常支持比 CNF 更通用的输入格式 [35]。

鉴于这些进展，从更一般的矩阵范式出发研究 QSAT 的复杂性是很自然的，同时这些范式仍需支持合适的图表示。一个显而易见的方向是将 CNF 与 DNF 结合起来，即由一个 CNF 公式和一个 DNF 公式构成合取。这种组合形式“CDNF”被 QBF 回溯搜索算法采用，因为它能够模拟电路级推理，同时利用经过优化的数据结构 [31]。

由于合取析取范式是合取范式的严格推广，我们在第 IV 节针对合取范式的关联图所建立的下界可以直接推广。然而，与合取范式的情形不同，我们的归约还直接排除了：对于合取析取范式中的矩阵，关于原始图的树深度和反馈顶点数参数，QSAT 具有固定参数可处理性的可能。

3. 缩小原始图上的差距。对于经典合取范式矩阵，第 IV 节中的 SAW 归约几乎——但尚未完全——弥合了前述原始图树宽与顶点覆盖数之间的复杂性理论差距。具体而言，我们证明，对于树深度或反馈顶点

数有界的实例，在原始图中增加一个子句就会导致不可解性。鉴于这一进展，我们认为，确定 QSAT 关于这两个参数的参数化复杂性，是理解该问题复杂性图景时尚待解决的主要开放问题。

作为最后一项贡献，我们针对 QSAT 提出了新的固定参数算法，旨在缩小这一差距。首先，我们为以原始图的反馈边数为参数的 QSAT 得到了一个线性核（因此也得到了一个固定参数算法）。其次，我们证明了该问题相对于顶点覆盖数的若干种放宽参数具有固定参数可处理性；这些放宽参数可视为通向合取范式原始图树深度的“垫脚石”。图 2（左）更详细地概述了我们的结果。

B. 方法与技术

为了在上述图表示中为介于树宽与顶点覆盖数之间的参数建立精细下界，我们采用结构感知（SAW）归约的概念，如图 2（右）所示。我们构造了具体的 SAW 归约；从概念上说，它们是从 QSAT 到 QSAT 的自归约，通过牺牲反馈顶点数或树深度参数的指数级缩减，换取对相应参数的运行时间依赖呈指数级增长。为得到紧确的下界，理想情况下使其与现有上界相匹配（并排除显著优于树宽算法的算法），必须谨慎权衡，使运行时间依赖增长的量级不超过参数缩减的量级。更具体地说，我们的变换从以相应参数 k 和量词深度 ℓ为参数的 QSAT 归约到以 $\log(k)$ 为参数、量词深度为 $\ell+1$ 的 QSAT。通过迭代这一构造（另见图 2（右）），我们以 i 的量词深度增长为代价，将参数缩减 i 重指数级（从 k 缩减至 $\log^i(k)$）；随后，在指数时间假设成立的前提下，求解该量化布尔公式需要关于 $\log^i(k)$ 的 $\ell+i$ 重指数级时间。

作为我们归约的推论，我们还得到了一个关于经典复杂性、有趣的结果：事实证明，只需额外增加一个子句，就足以使 QSAT 在众所周知的可处理片段——2-CNF 公式上变得不可解。更具体地说，量词深度为 $\ell > 1$ 的 2-CNF 加一个子句上的 QSAT 确实是 $\Sigma^P_{\ell-1}$-完全的，见推论 IV.4。

值得注意的是，我们的归约构造还使我们能够将已确立的下界加强到完全限制于最内层量词（块）变量的图表示。这是因为我们具体的 SAW 归约经过设计，使得大多数结构依赖关系都位于所构造实例的最内层量词块中。此外，对于涵盖到（几乎）简单路径的顶点删除距离的参数，以及树深度的受限变体，这些下界同样成立。这两项结果是特定于构造的推论，但这些发现实际上明显强于关于反馈顶点集和树深度的下界，从而为 QSAT 的困难性提供了更深入的认识。

利用 SAW 归约得到的下界结果，使我们能够较为全面地描绘关联图的（精细粒度）参数化复杂性：我们将已有的困难性结果加强到了反馈顶点集和树深度等限制更严格的参数。对于关联图，我们通过给出 CQSAT（即限制为合取范式公式的 QSAT）关于顶点覆盖数和反馈边数的固定参数可处理算法，补充了这些否定性结果。这里的主要技术贡献是针对原图和关联图的反馈边集提出了一种核化算法。

接下来，我们将目光转向利用原始图的结构限制来求解 CQSAT。虽然我们仍无法确定，以原始图的反馈顶点数或树深度为参数时，CQSAT 是否固定参数可处理，但我们在证明后者具有固定参数可处理性方面取得了一些进展。具体而言，借助对欣蒂卡博弈中获胜策略的新见解，我们为所谓的 c-删除集参数的三个变体设计了固定参数算法。该参数介于顶点覆盖数与树深度之间。

II. 预备知识

我们假设读者掌握图论基础知识，参见 [19], [9]。如果满足 $V \subseteq V'$, $E \subseteq E'$，则图 $G = (V, E)$ 是图 $G' = (V', E')$ 的子图。图的（连通）分量是极大的连通子图。如果图中不存在构成环的子图，则该图是无环的。对于图 $G = (V, E)$ 和由顶点（边）组成的集合 $S \subseteq V$ ($D \subseteq E$)，我们定义从 G 中通过 $G - S := (V \setminus S, \{e \mid e \in E, e \cap S = \emptyset\})$（通过 $G - D := (V, E \setminus D)$）得到的差图。此外，给定图 $G_1 = (V_1, E_1)$ 和 $G_2 = (V_2, E_2)$，它们的并定义为 $G_1 \sqcup G_2 := (V_1 \cup V_2, E_1 \cup E_2)$。表达式 $\mathrm{tow}(\ell, k)$ 表示一个底数为 2、高度为 ℓ 的指数塔，顶层为 k。

计算复杂性：我们简要介绍参数化复杂性 [27], [44]。设 Σ 和 Σ' 是两个有限的非空字母表。参数化问题 L 是某个有限字母表 Σ 上 $\Sigma^* \times \mathbb{N}$ 的一个子集。若存在一个可计算函数 f 和一个算法，能够在固定参数可处理时间内判定 $(\mathcal{I}, k) \in L$，则称 L 是固定参数可处理的（fpt）。

前缀[A]	矩阵	矩阵范式	图	复杂性	Ref
		CNF		多项式空间	[59]
qd ℓ		CNF		Σ_ℓ^P / Π_ℓ^P	[59]
qd ℓ		2,1-合取析取范式		$\Sigma_{\ell-1}^P$ / $\Pi_{\ell-1}^P$	推论 IV.4
	tw	CNF	P	多项式空间	[46], [3]
qd	tw	CNF	P, I	$\triangle^\dagger$ / $\triangledown^\ddagger$	命题 II.3 [12],[10] / [46],命题 II.4 [26]
qd	tw	m,1-合取析取范式[B]	P	$\blacktriangle^\dagger$ / $\triangledown^\ddagger$	定理 III.2 / 命题 II.4 [26]
qd	fvs	m,1-合取析取范式[B]	P	$\blacktriangle^\dagger$ / $\blacktriangledown^\ddagger$	推论 III.4 / 定理 IV.5
qd	fvs	CNF	I	$\blacktriangle^\dagger$, $\blacktriangledown^\ddagger$	推论 III.4 / 推论 IV.6
qd	td	m,1-合取析取范式[B]	P	$\blacktriangle^\dagger$ / $\blacktriangledown^{\sim\ddagger}$	推论 III.4 / 定理 IV.7, 推论 IV.8
qd	td	CNF	I	$\blacktriangle^\dagger$ / $\blacktriangledown^{\sim\ddagger}$	推论 III.4 / 推论 IV.9
	fes	CNF	P	■	定理 V.2
	fes	CNF	I	■	定理 V.5
	dels	CNF	P	■[C]	定理 VI.3 , 定理 VI.5 , 定理 VI.7
	vc	m,1-合取析取范式[B]	P	$\blacktriangle^{\dagger_2}$ / $\triangledown^{\ddagger_2}$	定理 III.1 / 命题 II.5 [38]
	vc	CNF	I	$\blacktriangle^{\dagger_2}$	定理 V.1

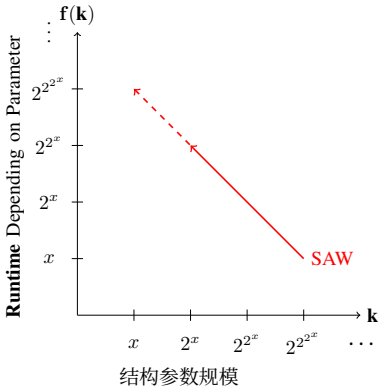


图 2：（左）：当以 QBF Q 中前缀和矩阵所列参数进行参数化时，QSAT 的运行时间界。三角形 ▲ 表示已确立的精确上界，▼ 表示精确下界。▽ 和△ 分别表示此前已知的精确上界和下界。方框 ■ 表示新的固定参数可处理结果。图参数分别应用于原始图（P）或关联图（I）；其中：“qd”表示量词深度；“fvs”表示反馈顶点数；“td”表示树深度；“fes”表示反馈边数；“dels”表示大小+c 的 c-删除集。运行时间界通过标记缩写，其中 ℓ 表示前缀，k 表示矩阵的参数化。详细结果：†: $\mathrm{tow}(\ell, O(k)) \cdot \mathrm{poly}(|\mathrm{var}(Q)|)$；‡: $\mathrm{tow}(\ell, o(k)) \cdot \mathrm{poly}(|\mathrm{var}(Q)|)$；$^{\sim\ddagger}$: $\mathrm{tow}(\ell, o(k-\ell)) \cdot \mathrm{poly}(|\mathrm{var}(Q)|)$ ^{†_2}: $2^{2^{\mathcal{O}(k)}} \cdot \mathrm{poly}(|\mathrm{var}(Q)|)$，对于常数 m: $2^{k^{\mathcal{O}(m)}} \cdot \mathrm{poly}(|\mathrm{var}(Q)|)$；^{‡_2}: $2^{2^{o(k)}} \cdot \mathrm{poly}(|\mathrm{var}(Q)|)$；以及 ^{A}: ∃奇/∀偶。^{B}: 下界对 3,1-CDNF 已成立；^{C}: 在限制条件下固定参数可处理，并以 $|D|+c$ 作为 c-删除集的参数 D。（右）：结构感知归约，通过运行时间依赖来补偿指数参数下降。

$\mathcal{O}(f(k)\,\mathrm{poly}(\|\mathcal{I}\|))$，其中 $\|\mathcal{I}\|$ 是 $\mathcal{I}$ 的大小。设 $L \subseteq \Sigma^* \times \mathbb{N}$ 和 $L' \subseteq \Sigma'^* \times \mathbb{N}$ 是两个参数化问题。从 L 到 L' 的多项式时间参数化归约 r（简称 pp-归约）是从 $\Sigma^* \times \mathbb{N}$ 到 $\Sigma'^* \times \mathbb{N}$ 的多对一归约，满足：$(\mathcal{I}, k) \in L$ 当且仅当 $r(\mathcal{I}, k) = (\mathcal{I}', k') \in L'$，其中 $k' \leq p(k)$ 对某个固定的可计算函数 p 成立：$\mathbb{N} \to \mathbb{N}$，且 r 可在 $\mathcal{O}(\mathrm{poly}(\|\mathcal{I}\|))$ 时间内计算。参数值通常根据实例的某个结构性质 K 计算，该性质称为参数，例如最小反馈顶点集的大小或树宽。算法通常需要结构表示，而非参数值，即反馈顶点集或树分解。因此，设 Γ 是一个有限的非空字母表，并称 $S \in \Gamma^*$ 为 $\mathcal{I}$ 的结构表示。于是，参数 K 的参数化 κ 是一个映射 $\kappa: \Gamma^* \to \mathbb{N}$，它能在多项式时间内计算 $k = \kappa(S)$。

量化布尔公式（QBF）：布尔公式按通常方式定义 [7], [36]；文字是变量或其否定。我们令文字 l 的符号 $= 1$ 定义为：若 l 是变量，则 $\mathrm{sgn}(l) := 1$；否则 $\mathrm{sgn}(l) := 0$。对于布尔公式 F，我们用 $\mathrm{var}(F)$ 表示 F 中变量的集合。逻辑运算符 $\to$、$\wedge$、$\vee$、$\neg$ 分别指蕴含、合取、析取和否定，其含义与通常定义相同。项或子句是文字的集合 S，分别解释为文字的合取或析取。

我们用 $\mathrm{var}(S)$ 表示出现在 S 中的变量集合；不失一般性地，假设 $|S| = |\mathrm{var}(S)|$。如果 F 是子句的合取，并且 F 是项的析取，则布尔公式 F 处于合取范式（CNF）中；如果 F 是项的析取，则 F 处于析取范式（DNF）中。在这两种情况下，我们分别用其子句集合或项集合来表示 F。如果 F 中的每个集合都至多包含 d 个文字，则称布尔公式处于 d-CNF 或 d-DNF 中。令 $\ell \geq 0$ 为整数。量化布尔公式 Q 具有如下形式：$\mathcal{Q}.F$，其中前缀为 $\mathcal{Q} = Q_1 V_1 . Q_2 V_2 . \cdots Q_\ell V_\ell$，量词 $Q_i \in \{\forall, \exists\}$ 作用于 $1 \leq i \leq \ell$，而 $Q_j \neq Q_{j+1}$ 作用于 $1 \leq j \leq \ell - 1$；其中 V_i 是互不相交且非空的布尔变量集合，并满足 $\mathrm{var}(Q) := \mathrm{var}(F) = \bigcup_{i=1}^{\ell} V_i$；$F$ 是一个布尔公式。如果 F 处于（c-）CNF 中，则 Q 是一个（c-）CQBF。我们称 ℓ 为 Q 的量词深度，并令 $\mathrm{matr}(Q) := F$。此外，我们将 Q 的变量记为 $\mathrm{var}(Q) := \mathrm{var}(F)$，并将存在（全称）变量记为 $\mathrm{var}^\exists(Q)$（$\mathrm{var}^\forall(Q)$），分别定义为 $\mathrm{var}^\exists(Q) := \bigcup_{1 \leq i \leq \ell, Q_i = \exists} V_i$（$\mathrm{var}^\forall(Q) := \bigcup_{1 \leq i \leq \ell\ Q_i = \forall} V_i$）。

赋值是从变量集合 X 到 $X \to \{0,1\}$ 的映射 α。给定一个布尔公式 F 以及针对 $\mathrm{var}(F)$ 的赋值 α。那么，对于 F 属于合取范式的情况，$F[\alpha]$ 是通过移除每个 $c \in F$而得到的布尔公式

其中，若 $\alpha(x) = 1$，则取 $x \in c$；若 $\alpha(x) = 0$，则取 $\neg x \in c$；并分别从每个剩余子句中移除 $\alpha(x) = 0$ 以及 $\alpha(x) = 1$ 时对应的 $c \in F$ 文字 x 和 $\neg x$。类似地，对于 F 属于析取范式的情况，0 和 1 的值互换。更一般地，对于一个矩阵由合取范式公式F_1 和析取范式公式 F_2的合取构成的量化布尔公式 Q，以及针对$X \subseteq \mathsf{var}(Q)$ 的赋值 α，我们将 $Q[\alpha]$ 定义为由Q 得到的量化布尔公式：从前缀中移除由 α 赋值的变量，将 F_1 替换为 $F_1[\alpha]$，并将 F_2 替换为 $F_2[\alpha]$。

对于给定的量化布尔公式 Q 和一个赋值 α，$Q[\alpha]$ 是由 Q 得到的量化布尔公式，其中由 α 映射的变量 x 会相应地从前面的量词中移除，并且 $\mathsf{matr}(Q[\alpha]) = (\mathsf{matr}(Q))[\alpha]$。

如果存在一个针对 $\mathsf{var}(F)$ 的赋值 α，使得当 F 属于合取范式时 $F[\alpha] = \emptyset$，或者当 F 属于析取范式时 $F[\alpha] = \{\emptyset\}$，则布尔公式 F 求值为真（或称为满足）。我们称此时 α 满足 F，或者称 α 是 F 的满足赋值。如果 $\ell = 0$ 且 $\mathsf{matr}(Q)$ 在空赋值下求值为真，则量化布尔公式 Q 求值为真（或称为有效）。否则，即当 $\ell \neq 0$ 时，我们根据 Q_1 进行区分。如果 $Q_1 = \exists$，则当且仅当存在一个赋值 $\alpha\colon\ V_1 \to \{0,1\}$，使得 $Q[\alpha]$ 求值为真时，Q 求值为真。如果 $Q_1 = \forall$，则对于每个赋值 $\alpha\colon\ V_1 \to \{0,1\}$，$Q$ 求值为真，当且仅当 $Q[\alpha]$ 求值为真。若一个量化布尔公式求值为真时另一个也求值为真，反之亦然，则称这两个量化布尔公式等价。给定一个（C）QBF Q，量化布尔公式的求值问题 QSAT（CQSAT）询问 Q 是否求值为真。因此，SAT 是 QSAT 的一种情形，但限制为只有一个 $\exists$ 量词。一般而言，QSAT 是 PSPACE 完全的 [36], [47], [59]。

例 II.1。考虑合取范式量化布尔公式 $Q = \forall a, b. \exists c, d. C$，其中 $C := \{c_1, c_2, c_3, c_4\}$ 是由子句构成的合取，包含 $c_1 := \neg a \vee \neg b \vee c$, $c_2 := a \vee b \vee c$, $c_3 := \neg a \vee \neg c \vee d$ 和 $c_4 := a \vee \neg c \vee \neg d$。注意，$Q[\alpha]$ 在任意 $\alpha\colon \{a,b\} \to \{0,1\}$ 下均有效，这可以通过为任意 α 给出赋值 $\beta\colon \{c,d\} \to \{0,1\}$ 来证明。具体而言，令 $\beta(c) := 1$，当 $\alpha(a) = \alpha(b)$ 时；否则令 $\beta(c) := 0$。此外，令 $\beta(d) := 1$，当 $\alpha(a) = \alpha(b) = 1$ 时；否则令 $\beta(d) := 0$。事实上，对于任何这样的 α，我们都有 $C[\alpha][\beta] = \emptyset$ 和 $D[\alpha][\beta] = \{\emptyset\}$。例如，考虑满足 c_1, c_2 和 c_3 的 $\alpha = \{a \mapsto 0, b \mapsto 1\}$；那么 c_4 由 β 满足。■

矩阵公式的扩展正规化（CDNF）：我们的研究也探讨了量化布尔公式的一种自然且更一般的合取范式／析取范式（CDNF）。若一个最内层量词为 Q_ℓ 的量化布尔公式 Q 满足以下条件，则称其为 CDNF：对于合取范式 C 和析取范式 D，当 $Q_\ell = \exists$ 时，$\mathsf{matr}(Q) = C \wedge D$；当 $Q_\ell = \forall$ 时，$\mathsf{matr}(Q) = D \vee C$。自然地，若 C 属于 d-合取范式且 D 属于 d-析取范式，则称 Q 属于 d-CDNF。此外，若 C 属于 d-合取范式且 D 属于 1-DNF（即 D 是一个长子句，解释为

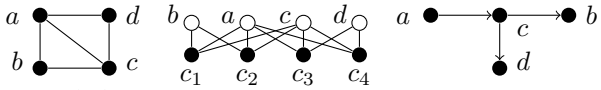


图 3：（左）：示例 II.1 中量化布尔公式 Q 的原始图表示 G_Q。（中）：量化布尔公式 Q 的关联图 I_Q。（右）：G_Q 的树深度分解。

作为单元素项的析取）。那么，问题 QSAT_ℓ 指的是限制为合取析取范式且量词深度为 ℓ 的量化布尔公式的 QSAT。

图表示：为了将图参数应用于（量化）布尔公式，我们需要一种图表示。对于合取范式或析取范式中的布尔公式 F，我们定义其原始图 $G_F := (\mathsf{var}(F), E)$ [53]，该图以 F 的变量为顶点；若两个变量同时出现在 F 的至少一个子句或项中，则在它们之间连一条边，即 $E := \{\{x,y\} \mid f \in F, \{x,y\} \subseteq \mathsf{var}(f)\ x \neq y\}$。公式 F 的关联图 $I_F := (\mathsf{var}(F) \cup F, E')$ 以公式中的变量和子句（或项）为顶点，并且 $E' := \{\{f,x\} \mid f \in F, x \in \mathsf{var}(f)\}$。对于一个处于合取析取范式中的量化布尔公式 Q，若 $\mathsf{matr}(Q)$ 为 $= C \wedge D$ 或 $\mathsf{matr}(Q)$ 为 $= D \vee C$，则分别令 Q 的原始图为 $G_Q := G_C \sqcup G_D$，并令 Q 的关联图为 $I_Q := I_C \sqcup I_D$。

例 II.2. 回顾例 II.1 中的 Q 和 $C = \mathsf{matr}(Q)$；观察图 3（左、中）中的原始图和关联图 G_Q、I_Q。假设一个由 Q得到的合取析取范式量化布尔公式 Q'，其中 $\mathsf{matr}(Q') := C \wedge D$，且 D 是（单元素）项的析取，即 $D := \{\{b\}, \{\neg d\}\}$。注意，根据定义，(1-)析取范式公式 D 不会在原始图 $G_{Q'}$中引入额外的边，即该图等价于不含 D的原始图 G_C。因此，G_Q 与原始图 $G_{Q'}$重合。■

树宽与路径宽度：设 $G = (V, E)$ 是一个图。图 G 的树分解（TD）[51], [52] 是一个二元组 $\mathcal{T} = (T, \chi)$，其中 T 是一棵树，χ 是一个映射，为 T的每个节点 t 分配一个集合 $\chi(t) \subseteq V$，称为一个包，并满足以下条件：(i) $_T\chi(t)$的 $V = \bigcup_t$和 $_T\{\{u,v\} \mid u,v \in \chi(t)$的 $E \subseteq \bigcup_t\}$；以及 (ii) 对于每个 q、s、t，如果 s位于从 q到 t的路径上，则有 $\chi(q) \cap \chi(t) \subseteq \chi(s)$。于是，$\mathsf{width}(\mathcal{T}) := \max_t$ of $_T|\chi(t)| - 1$。图 G 的树宽 $\mathsf{tw}(G)$ 是所有关于 G的树分解 $\mathcal{T}$ 的 $\mathsf{width}(\mathcal{T})$ 的最小值。对于 QSAT，已知以下可处理性结果。

命题 II.3（树宽上界 [12]）。给定任意量词深度为 ℓ 且具有 $k = \mathsf{tw}(G_Q)$ 的合取范式量化布尔公式Q，QSAT 在 Q 上可在时间 $\mathsf{tow}(\ell, \mathcal{O}(k)) \cdot \mathsf{poly}(|\mathsf{var}(Q)|)$ 内判定。

然而，人们并不认为这一结果能够得到显著改善，因为即使对于较弱的路径宽度，也存在限制。图 G 的路径宽度 $\mathsf{pw}(G)$ 是其所有树为路径的 G 的 TD 的最小宽度。

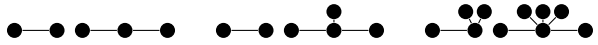


图 4：不相交路径（左）；半梯子图（中）；以及毛毛虫图（右）。

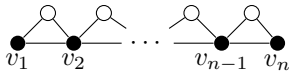


图 5：反馈顶点数为 $\mathcal{O}(n)$ 的路径宽度为 2 的图。

命题 II.4（路径宽度下界 [26]）。给定任意一个量词深度为 ℓ 且具有 $k = \text{pw}(G_Q)$ 的 CQBF Q 。那么，在指数时间假设成立的条件下，无法在 $\text{tow}(\ell, o(k)) \cdot \text{poly}(|\text{var}(Q)|)$ 时间内判定 Q 上的 QSAT 。

树深度：给定图 $G = (V, E)$。那么， G 的树深度分解 $T = (V, F)$是一个根树森林，对于每条边 $\{u, v\} \in E$ ，我们要求 u 是 T 中 v 的祖先或后代。 G 的树深度 $\text{td}(G)$ 是 G 的所有树深度分解中高度最小者，参见图 3（右）。

顶点覆盖数：给定图 $G = (V, E)$。那么，顶点集合 $S \subseteq V$ 是（ G 的）顶点覆盖，当且仅当对于每条边 $e \in E$ ，都有 $e \cap S \neq \emptyset$。此外，我们将图 G 的顶点覆盖数定义为 G 的所有顶点覆盖中规模最小者。有趣的是，以该数为参数时，QSAT 是可解的。

命题 II.5（顶点覆盖数的上界 [38]）。给定任意合取范式量化布尔公式 Q ，其 QSAT 的顶点覆盖数为 k ，其中 G_Q。那么，可以在时间 $2^{2^{\mathcal{O}(k)}} \cdot \text{poly}(|\text{var}(Q)|)$ ($2^{\mathcal{O}(k^3)} \cdot \text{poly}(|\text{var}(Q)|)$ 内判定 Q 的有效性（对于 3-CNF 中的 $\text{matr}(Q)$ ）。

反馈集与距离度量：针对以树宽（路径宽度）或顶点覆盖数为参数的 QSAT 的下界结果促使我们考虑其他参数。

设 $G = (V, E)$ 为一个图。那么，顶点集 $S \subseteq V$ 称为 G 的反馈顶点集（ FVS ），如果 $G - S$ 是无环图； G 的反馈顶点数指 G 的所有反馈顶点集中最小的大小。此外，如果 $G - S$ 是半梯（图），由彼此（顶点）不相交的路径组成，并且每个顶点至多与一个新顶点相邻，则称 S 为到半梯的距离集。如果允许每个顶点与多个这样的新顶点相邻，我们就称该图为毛毛虫图，参见图 4。这些距离集中 S 的最小 $k = |S|$ 即为距离（到相应图类）。对于某个整数 c，如果 $G-S$ 的每个分量都至多包含 c 个顶点，则称 S 为一个 c-删除集。若 $G - D$ 是无环图，则称边集 $D \subseteq E$ 为 G 的反馈边集（FES）。

我们利用这些集合 S、 D 来处理量化布尔公式 Q，其中$G = G_Q$。那么，如果对于任意两个不同的顶点 u、 v，在 $G_Q - S$ 中至多存在一个子句或项 f属于 $\text{matr}(Q)$，且满足 u、 $v \in \text{var}(f)$，则称其是稀疏的。

示例 II.6。回顾示例 II.1 中的量化布尔公式 Q ，并注意到其反馈顶点数为 1，例如， $\{a\}$ 是 G_Q 的反馈顶点集，同时也是到 G_Q 的距离集。然而， Q 的稀疏反馈顶点数为 2，因为无法从 G_Q 中移除单个顶点，使得每条边至多对应 $\text{matr}(Q)$ 的一个子句。集合 $\{a, c\}$ 是 Q 的稀疏反馈顶点集，因为 $G_Q - \{a, c\}$ 不含边。对于 G_Q ，路径宽度与稀疏反馈顶点数相同；而图 5 的路径宽度为 2，但只能容纳一个较大的反馈顶点集，例如所有白色节点。■

受相关且更一般的参数层次结构研究的启发，[58], [16], 我们为量化布尔公式建立了参数层次结构：图 6 描绘了这些参数，其中从参数 p_1 指向参数 p_2 的有向弧表示 p_1 弱于 p_2，即 p_2 被 $\mathcal{O}(p_1)$ 上界约束。因此，关于较弱（线性更小）参数 p_1 的下界构成了更强的结果，并且会自动推广到更强的参数 p_2。

例 II.7. 注意，即使对于一个合取范式量化布尔公式 Q， k_G 的反馈顶点数 G_Q 与 I_Q 的反馈顶点数 k_I 也不可比较，参见图 6。显然，通过构造一个含有一个大型子句的实例即可得到 $k_I \ll k_G$ 。不过，也存在 $k_I \gg k_G$的情况：可以构造变量对，使每对变量都出现在（至少）两个大小为 2 的子句中，即每对变量都参与了 G_I中的一个环。此时， k_G 为零，而 k_I 等于变量对的数量。

III. 结构感知（SAW）归约

回顾一下，使用树宽（路径宽度，参见命题 II.4）求解 QSAT 的运行时间，与以顶点覆盖数为参数求解 QSAT 的运行时间之间存在差距（参见命题 II.5）。有趣的是，针对合取范式中 QSAT 和顶点覆盖数的运行时间界，同样适用于合取析取范式。

定理 III.1（QSAT_ℓ 与顶点覆盖数的上界， $\star$[3]）。存在一种算法，对于给定的顶点覆盖数为 k G_Q的合取析取范式量化布尔公式 Q，可在时间 $2^{2^{\mathcal{O}(k)}} \cdot \text{poly}(|\text{var}(Q)|)$内判定 Q 是否为真。如果Q 属于 d-合取析取范式，该算法的运行时间为 $2^{k^{\mathcal{O}(d)}} \cdot \text{poly}(|\text{var}(Q)|)$。

证明（思路）。通过为 DPLL 风格回溯算法加入公式缓存，可以建立该结果 [5]。通过赋值变量得到的矩阵子公式数量，可以限制为关于变量数量呈线性、而关于顶点覆盖大小仅呈指数的函数。

[3]带星号（“$\star$”）标记的陈述在附录 中证明。

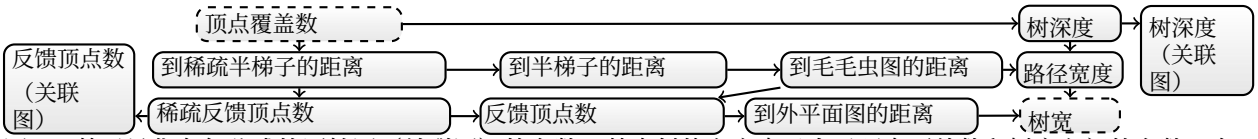


图 6：基于量化布尔公式的原始图（关联图）的参数，其中斜体文字表示介于顶点覆盖数和树宽之间的参数。有向弧表示源参数弱于目标参数，即目标参数受源参数线性有界。粗体参数标记了针对 QSAT$_\ell$ 选出的新的下界。

这一上界限制了搜索树的大小。 □

同样，关于树宽的已知运行时间结果（命题 II.3）也适用于合取析取范式。

定理 III.2（QSAT$_\ell$ 与树宽的上界）。给定任意量词深度为 ℓ 且 k =tw(G_Q) 的合取析取范式量化布尔公式 Q 。则可在时间 tow($\ell, \mathcal{O}(k)$) · poly(|var(Q)|) 内判定 Q 上的 QSAT$_\ell$ 。

我们通过依赖以下结构感知归约 [26], [24]的概念来证明这一结果。这些归约将成为本文新下界结果构造中的关键分量。它们提供了一种构造性方法，利用实例的实际结构表示，从而根据该表示精确界定参数的增加（或减少）。

定义 III.3（SAW 归约）。设 Σ、 Σ'、 Γ、 Γ' 为字母表，$\mathcal{P} \subseteq \Sigma^* \times \mathbb{N}$、 $\mathcal{P}' \subseteq \Sigma'^* \times \mathbb{N}$ 为具有参数化 κ、κ' 的参数化问题， f 为可计算函数。若一个 f-结构感知归约 $\mathcal{R}$ 从 $\mathcal{P}$ 到 $\mathcal{P}'$ 将 $\Sigma^* \times \Gamma^*$ 映射到 $\Sigma'^* \times \Gamma'^*$，且对于满足 $(\mathcal{I}', S') = \mathcal{R}(\mathcal{I}, S)$ 的 $(\mathcal{I}, S) \in \Sigma^* \times \Gamma^*$，有 $((\mathcal{I}, S), \kappa(S)) \mapsto ((\mathcal{I}', S'), \kappa'(S'))$ 是一个满足以下条件的 pp-归约：(i) $S' = g(S)$ 对于多项式时间函数 g （函数依赖）；(ii) $\kappa'(S') \leq \mathcal{O}(f(\kappa(S)))$（$f$-有界性）。

SAW 归约的定义有以下作用。首先，这种归约总是会为约化实例提供一种结构化表示，由此，(i) 函数依赖能够立即让我们了解如何获得这种表示。此外，性质 (ii) f-有界性确保约化实例得到的参数满足精确的保证，这对下一小节至关重要。

为展示这些归约，我们简要解释图 6 中的弧。有趣的是，图 6 中几乎每条弧都可以通过平凡的线性 SAW 归约来证明：该归约接收一个 QSAT$_\ell$ 的量化布尔公式 Q 以及相应参数的结构表示 S，并返回 (Q, S)。事实上，例如，任意顶点覆盖 S 都是到半梯子的距离集，并且对于任意路径分解 S，它都是一个树分解。此外，任意到外平面的距离集 S 可以转换为 TD S'，因为每个外平面图 [60] 都有一个宽度至多为 2 的 TD $\mathcal{T}$，而我们通过将 S 添加到 $\mathcal{T}$的每个包中来获得 TD S' 。

有了这些 SAW 归约，我们便可以轻松证明定理 III.2。由于树宽参数同时受到反馈顶点数和树深度的线性界定，我们可以立即得到以下结果。

推论 III.4（QSAT$_\ell$ 的上界：反馈顶点数/树深度）。给定任意一个量词深度为 ℓ 、其中 k 表示 G_Q 的反馈顶点数或树深度的合取析取范式量化布尔公式 Q 。则可在 tow($\ell, \mathcal{O}(k)$) · poly(|var(Q)|) 时间内判定 Q 上的 QSAT$_\ell$ 。

IV. 通过 SAW 归约得到的下界

为了为参数化问题建立条件下界，可以从 3-CNF 上的 SAT 归约，然后直接应用广为接受的指数时间假设（ETH） [34]。事实上，已有许多研究结果，其中约化实例的参数取决于布尔公式中的变量数，例如 [42], [38], [48]，后立即应用 ETH。通常，ETH 的推论已经足够：声称 3-CNF 上的 SAT 无法在 $2^{o(k)} \cdot \text{poly}(n)$ 时间内求解，其中 k 是实例的一个参数， n 是变量数。

在确立结构感知（SAW）归约这一概念后，我们将这种归约作为一种精确工具，用于推广命题 II.4 的下界结果。具体而言，下一小节着重定义一种从 QSAT 到 QSAT 的自归约，通过增加量词深度来换取所关注参数的指数级下降。通过这种方式，在假设指数时间假设（ETH）成立的前提下，我们能够证明与相应上界匹配的下界。

为了寻找合适的候选参数，回顾一下，对于顶点覆盖数 k，无论量词深度如何，QSAT 都可以在关于 k 的双指数运行时间内求解。然而，对于树宽（或路径宽度）而言，情况并非如此，因为当量词深度为 ℓ 时，所需运行时间是路径宽度的 ℓ 重指数，参见命题 II.4。这促使我们着手研究位于顶点覆盖数与路径宽度“之间”的合适参数。

在第 IV-A 节中，我们证明（稀疏）反馈顶点数同样是不充分的，即我们针对该参数得到了与推论 III.4 的上界相匹配的 ETH 紧下界。第 IV-B 节对该归约进行了调整，从而进一步深入揭示了树深度问题的困难性。此外，本节还针对具有高树宽的实例，给出了一个类似于推论 III.4 的树深度 ETH 紧下界。

A. 反馈顶点数的紧 QBF 下界

整体方法通过 SAW 归约进行如下。我们假设有一个 QSAT$_\ell$ 实例 Q 以及一个稀疏反馈顶点集 S，满足 $|S| = k$。然后，我们设计一个 log(k)-SAW 归约 $\mathcal{R}$，构造一个等价的量化布尔公式 Q'，使得 Q' 具有一个大小为 $\mathcal{O}(\log(k))$ 的稀疏反馈顶点集 S'。

不失一般性地，我们将讨论范围限制在量化布尔公式 Q 的最内层量词为 $\exists$的情形，因为对另一种情形进行简单调整即可，或者可以求解逆问题，并在常数时间内对结果取逆。此外，我们假设 Q 的第一个量词同样为 $\exists$。设 $Q = \exists V_1.\forall V_2.\cdots\exists V_\ell.C \wedge D$为这样一个量化布尔公式，它具有一个稀疏反馈顶点集 S，满足 $k = |S|$。为了证明我们的下界，我们实际上在限制更严格的 3,1-CDNF 形式下证明一个更强的结果，其中假设 C 为 3-CNF，D 为 1-DNF，即$G_Q = G_C$，如例 II.2 中所述。最后，我们假设每个 $c_i \in C$ 恰好由三个文字组成；不过，该归约也适用于单个子句大小更小的情况。通过 SAW 归约得到的 Q' 的约化实例 Q' 和稀疏反馈顶点集 S' 利用额外的量词块来“展开” S'（即重构 S的一个赋值）。

辅助变量：为了构造 Q'，我们需要以下额外的（辅助）变量。首先，我们使用指针变量或索引变量，以便精确寻址 S 中的一个元素。为了在求值 3-合取范式（3-析取范式）公式时寻址 S 个元素，我们需要其中三个索引。这些索引变量的形式为 VarIdxs：$=\{idx_j^1,\ldots,idx_j^{\lceil\log(|S|)\rceil} \mid 1 \le j \le 3\}$，直观地说，对于三个索引中的每一个索引，它们都允许我们通过对 $\lceil\log(k)\rceil$ 个布尔变量进行特定赋值来“寻址” S 中的 k 个元素。这些每个索引对应的 $2^{\lceil\log(k)\rceil}$ 个变量组合 j 足以寻址 S 中的任意 k 个元素。为此，我们为每个元素 $x \in S$ 以及每个 $1 \le j \le 3$ 分配一个集合，该集合由索引变量 idx_j^1，.....，$idx_j^{\lceil\log(|S|)\rceil}$ 上文字的任意但固定且唯一的组合构成，记为 $[\![x]\!]_j$。

此外，对于每个子句 $c_i \in C$，其中包含 $c_i = \{l_1, l_2, l_3\}$，我们假设 c_i 的文字之间存在任意排序，并将 lit(c_i, j) := l_j 写作 c_i（$1 \le j \le 3$）中的第 j 个文字。我们还需要三个布尔变量 val_1、val_2、val_3，其中 val_j 保存通过第 j 个索引对应的变量寻址的 S 元素的真值（索引）值。这些变量称为 VarVals := $\{val_1, val_2, val_3\}$。最后，我们使用一个变量存储 D 是否得到满足，以及 $|C|$ 个辅助变量来指示子句 $c \in C$ 是否得到满足。这些变量由可满足性变量集合 VarSat : $= \{sat, sat_1, \ldots, sat_{|C|}\}$ 寻址，其中我们假设子句 C=$\{c_1, \ldots, c_{|C|}\}$ 按照某个固定的全序排列。

归约：该归约 $\mathcal{R}$ 接收 Q 和 S，并构造一个实例 Q' 以及一个稀疏反馈顶点集 S'，它们属于 Q'。量化布尔公式 Q' 的形式为 Q' : $=\exists V_1.\forall V_2.\cdots\exists V_\ell.\forall$VarIdxs、VarVals、VarSat$.C' \vee D'$，其中 C' 为析取范式，定义为若干项的析取：

$$x \wedge \bigwedge_{b \in [\![x]\!]_j} b \wedge \neg val_j \quad \text{for each } x \in S, 1 \le j \le 3 \tag{1}$$

$$\neg x \wedge \bigwedge_{b \in [\![x]\!]_j} b \wedge val_j \quad \text{for each } x \in S, 1 \le j \le 3 \tag{2}$$

$sat_i \wedge l$ 对于每个 $c_i \in C$、$1 \le j \le 3$，满足 lit(c_i, j) $= l$, var(l) $\in$ var(C) $\setminus S$ (3)

$sat_i \wedge \neg b$ 对于每个 $c_i \in C$、$1 \le j \le 3$、$x \in S$, $b \in [\![x]\!]_j$ 满足 var(lit(c_i, j)) $= x$ (4)

$sat_i \wedge val_j$ 对于每个 $c_i \in C$、$1 \le j \le 3$、$x \in S$，满足 lit(c_i, j) $= x$ (5)

$sat_i \wedge \neg val_j$ 对于每个 $c_i \in C, 1 \le j \le 3, x \in S$，满足 lit($c_i$, j) $= \neg x$ (6)

$sat \wedge l$ for each $\{l\} \in D$ (7)

此外，我们将 D' 定义为 1-CNF，它是以下单子句的合取。

$$\neg sat_i \quad \text{for each } 1 \le i \le |C| \tag{8}$$

$$\neg sat \tag{9}$$

注意，新的辅助变量出现在 Q' 的最内层全称量词之下。因此，直观地说，公式 (1) 确保每当某个 $x \in S$ 被赋值为 1，且第 j 个索引指向 x 时，如果 val_j 被赋值为 0，我们就会“跳过”相应的赋值。这类似于 $x \in S$ 被赋值为假时的公式 (2)，它确保对于 Q' 的其余公式，每当第 j 个索引指向某个 $x \in S$ 时，相应的值 val_j 与 x 的隐式赋值一致。

公式（3）–（6）用于检查子句 $c_i \in C$ 是否得到满足。直观地说，变量 sat_i充当开关：若将其设为真，就要求子句 c_i 得到满足。公式（3）确保当文字 $l \in c_i$（其变量 var(l) 不在 S 中）被赋值为真时，c_i 得到满足。对于变量 var(l) 属于 S 的文字 $l \in c_i$，如果对应的第 j 个索引（j，满足 $l =$ lit(c_i, j)）不指向 var(l)，那么公式（4）为真；否则，如果目标文字为真，则公式（5）和（6）中的一个为真。最后，公式（7）中的一个为真时，D 也为真，这与公式（3）的情况类似。注意，由于 VarSat 变量

是全称量化的，多个 sat_i 变量可能会被设为真。直观地说，这会让公式（3）–（7）中的某一项更“容易”满足，因为只需满足子句 c_i 中的一个即可。VarSat 变量唯一有问题的赋值是将 sat 以及所有 sat_i 变量都设为假。1-CNF 公式 D'（即公式（8）、（9））可避免这种情况。

例 XI.1（$\star$）在一个具体公式上阐释了 $\mathcal{R}$。结构感知性：除了 Q'之外，上述归约 $\mathcal{R}$还进一步产生了由 S' 定义的 Q' 的稀疏反馈顶点集：$=$ VarIdxs $\cup$VarVals $\cup\{sat\}$。事实上，与 $|S|$ 相比，S' 的规模呈指数级缩小，因此 $\mathcal{R}$ 确实是一种结构感知归约。该归约以及 Q与 Q'、S 与 S' 之间的关系如图 7 所示。形式上，我们得到以下结果，说明 $\mathcal{R}$ 确实是针对稀疏反馈顶点集的一种 SAW 归约。

引理 IV.1（降低反馈顶点数，$\star$）。给定一个属于 3,1-CDNF 的量化布尔公式 Q 以及 Q的稀疏反馈顶点集 S，$\mathcal{R}$ 构造出一个量化布尔公式 Q'，其具有 Q' 的稀疏反馈顶点集 S'，使得 $|S'|$ 属于 $\mathcal{O}(\log(|S|))$。

关于 C'的 3-析取范式：注意到，由归约 $\mathcal{R}$ 生成的公式 C'几乎已经是 3-析取范式。唯一尚未采用所需格式的公式是公式 (1) 和 (2)。不过很容易观察到，即使是这些公式也可以进行转换，使每个项至多使用 3 个文字。为此，需要引入额外的辅助变量（将其添加到最内层 $\forall$ 量词中）。实际上，一种直接的转换方法是递归地将公式 (1) 和 (2) 拆分为两个项：第一个项由两个文字和一个新辅助变量 v组成，并将该变量以正形式添加；第二个项则由该项的其余文字以及 $\neg v$组成。相应地，每个项至多包含两个新的辅助变量，并且只需注意，包含 $x \in S$ 或 $\neg x \in S$ 的结果项不能使用这两个辅助变量，以防止结果原始图中出现环。

运行时间与正确性：接下来，我们展示 $\mathcal{R}$ 的运行时间与正确性，然后给出主要结果。

定理 IV.2（运行时间，$\star$）。对于一个 3,1-CDNF 形式的量化布尔公式 Q，其 matr(Q) 为 $=C\wedge D$，且变量集合 var(Q) 为 $S\subseteq$，Q 的运行时间为 $\mathcal{R}$ $\mathcal{O}(\lceil\log(|S|+1)\rceil\cdot(|S|+|C|)+|D|)$。

定理 IV.3（正确性，$\star$）。给定一个 3,1-CDNF 中的量化布尔公式 Q，以及 Q 的变量集合 $S\subseteq$ var(Q)，归约 $\mathcal{R}$ 会计算出一个与 Q 等价的实例 Q'。事实上，任意对 matr(Q) 中变量的赋值 α 满足 matr(Q)，当且仅当 α 扩展 α' 到变量 VarIdxs $\cup$VarVals $\cup$VarSat 上的每一种方式都满足 matr(Q')。

众所周知，将 QSAT 限制为 2-CNF 公式 [2] 时，可在多项式时间内求解。作为我们归约 $\mathcal{R}$ 的一项应用，我们现在指出，即使只允许一个任意长度的子句，也会导致不可处理性。

推论 IV.4（$\star$）。对于一个量词深度为 $\ell\geq 2$ 的量化布尔公式Q=$Q_1V_1\ \cdots Q_\ell V_\ell.C\wedge D$，若 $Q_\ell=\exists$、C 属于 2-CNF，且 D 属于 1-DNF，则问题 QSAT 是 $\Sigma^{\mathrm{P}}_{\ell-1}$-完全的（若 $Q_1=\exists$、ℓ 为奇数），且是 $\Pi^{\mathrm{P}}_{\ell-1}$-完全的（若 $Q_1=\forall$、ℓ 为偶数）。

当最内层量词为全称量词时，也可以得到一个关于长项的类似结果。

下界结果：在确立了上述归约的结构感知性、运行时间以及正确性 $\mathcal{R}$ 之后，我们继续给出下界结果。

定理 IV.5（稀疏反馈顶点集的下界，$\star$）。给定一个量词深度为 ℓ、采用合取析取范式的任意量化布尔公式 Q，以及 Q的最小稀疏反馈顶点集 S，其中 $k=|S|$。那么，在指数时间假设下，无法在时间 tow($\ell,o(k)$) $\cdot$ poly(|var(Q)|) 内判定 Q 上的 QSAT$_\ell$。

因此，我们得到以下结果。

推论 IV.6（关联反馈顶点集的下界，$\star$）。给定一个任意的量化布尔公式 Q，其 Q具有 $F=$ matr(Q)，并且处于合取范式（析取范式）中，使得 Q的最内层量词 Q_ℓ 为 $Q_\ell=\exists$（$Q_\ell=\forall$），且 I_F 的反馈顶点数为 k。那么，在指数时间假设下，无法在时间 tow($\ell,o(k)$) $\cdot$ poly(|var(Q)|) 内判定 Q 上的 QSAT$_\ell$。

（稀疏）反馈顶点数的下界也适用于限制更严格的参数，详见附录。此外，即使将反馈顶点集限制为最内层量词块中的变量，该下界仍然成立，详见推论 X.3。

B. 树深度的困难性见解与新下界

截至目前，我们在第 IV-A 节讨论了为什么对于反馈顶点数而言，推论 III.4 无法得到显著改进。在本节中，我们针对树深度给出一个困难性结果，其形式为一个条件下界。值得注意的是，我们针对树深度的方法同样涉及 SAW 归约，其中我们甚至可以重用归约的主要部分 $\mathcal{R}$，该归约由公式 (1)–(9) 定义。设 $Q=\exists V_1.\forall V_2.\cdots\exists V_\ell.C\wedge D$ 是一个采用合取析取范式表示的量化布尔公式，并设 T 是 G_Q 的一个树深度分解，该分解由一条高度为 h的路径 S 组成，其中路径中的每个元素都可能连接到一棵高度为常数的树。对 Q和 S 应用 $\mathcal{R}$ 后的结果如图 8 所示。最后的规范化步骤（从 DNF 到 3-析取范式）由于在规范化公式 (1) 和 (2) 时引入了额外的辅助变量，导致出现多条长度为 $\mathcal{O}(\log(h))$ 的路径。这些路径并不会增加稀疏反馈顶点集的大小，因此此前可以忽略。可是，降低树深度需要压缩这些路径中的每一条，因此我们必须将以单个集合 $Q=\exists V_1.\forall V_2.\cdots\exists V_\ell.C\wedge D$ 为参数的归约 $\mathcal{R}$ 转变为一个能够同时处理多条路径的 SAW 归约

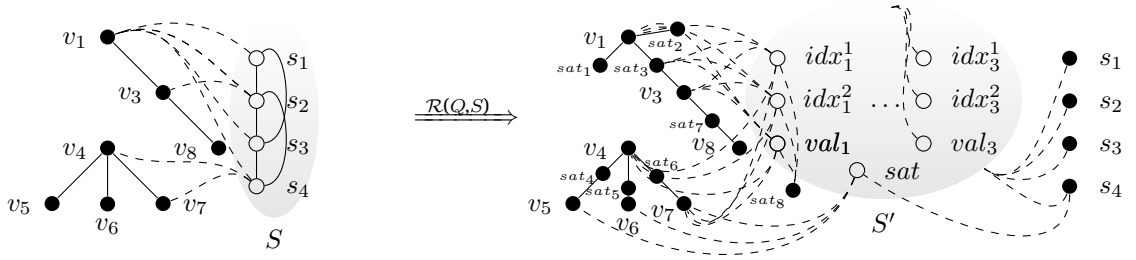


图7：针对某个量化布尔公式 Q 的结构感知归约 $\mathcal{R}$；虚线表示可能稠密的图部分。（左）：原始图 G_Q 及其连接 G_Q 的稀疏反馈顶点集 S。（右）：对应的原始图 $G_{Q'}$（简化后）以及由 $\mathcal{R}(Q,S)$ 得到的稀疏反馈顶点集 S'，其中 Q' 和 S' 由此获得。该示意图展示了 matr(Q) 的三种不同类型的子句：类型（i）仅使用 S 中的变量，例如 sat_8；类型（ii）使用 S 中的两个变量，例如 sat_1、sat_2；类型（iii）仅使用 S 中的一个变量，例如 sat_3、sat_4、sat_5、sat_6、sat_7。

不会增加稀疏反馈顶点集的大小，因此此前可以忽略。但降低树深度需要压缩这些路径中的每一条，因此我们必须将以单个集合 S 为参数的归约 $\mathcal{R}$ 转换为同时处理多条路径的 SAW 归约 $\mathcal{R}_{\mathsf{td}}$。$\mathcal{R}_{\mathsf{td}}$ 的形式化细节见附录。

利用 $\mathcal{R}_{\mathsf{td}}$，我们得到关于树深度 k 的以下下界；当量词深度 ℓ 满足 $k \in \mathcal{O}(\ell)$ 时，该下界达到指数时间假设意义下的紧致性，见推论 III.4。

定理 IV.7（树深度分解的下界，$\star$）。给定任意一个量词深度为 ℓ、属于合取析取范式的 QBF Q，以及一个高度为 $k=$ 的 G_Q 的树深度分解 T td(G_Q)。那么，在指数时间假设下，无法在时间 tow($\ell, o(k-\ell)$) $\cdot$ poly(|var(Q)|) 内判定 Q 上的 QSAT_ℓ。

我们证明，在指数时间假设下，该结果仍然蕴含一个运行时间层次，其中塔高与 QBF 的量词深度呈线性关系。

推论IV.8（树深度的下界，$\star$）。存在一个线性函数 f 和一个整数 $\ell_0 > 0$，使得对于任意一个量词深度为 $\ell \geq \ell_0$、属于 3,1-CDNF 的 QBF Q：在指数时间假设下，无法在时间 tow($f(\ell)$ o(td(G_Q))) $\cdot$ poly(|var(Q)|) 内判定 Q 上的 QSAT_ℓ。

推论IV.9（关联树深度的下界，$\star$）。给定一个 QBF Q，其 $F=$matr(Q) 位于合取范式（析取范式）中，且 Q 的最内层量词 Q_ℓ 为 $Q_\ell = \exists$（$Q_\ell = \forall$），并且 $k=$ td(I_Q)。那么，在指数时间假设下，无法在时间 tow($\ell, o(k-\ell)$) $\cdot$ poly(|var(Q)|) 内判定 Q 上的 QSAT_ℓ。

注意，该推论立即给出了一个类似于关联图上的推论 IV.8的结果。

V. 使用顶点覆盖和反馈边集的算法

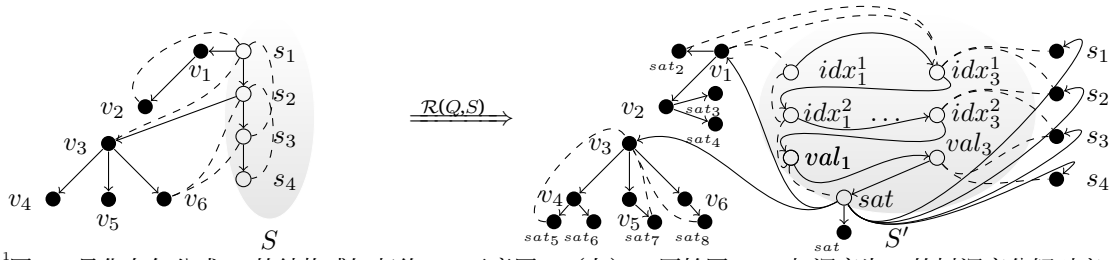
上一节的结果已经为考察关联图上许多最重要的结构参数时 CQSAT 的（精细化）参数化复杂性提供了相当全面的图景。特别是，这些结果排除了树深度和反馈顶点集的固定参数可处理性。在本节中，我们将通过给出以顶点覆盖数以及反馈边集数为参数的 CQSAT 固定参数可处理算法，来补充关联图的这一图景。我们首先介绍顶点覆盖数算法，该算法本质上基于以下简单观察：顶点覆盖数较小的公式不可能包含太多不同的子句；此外，还利用了众所周知的结果，即以子句数 [65] 为参数时，CQSAT 是固定参数可处理的。

定理 V.1（$\star$）。给定任意合取范式量化布尔公式 Q of QSAT，其中 k是 I_Q的顶点覆盖数。那么，Q 的有效性可以在时间 $1.709^{3^k} \cdot$poly(|var(Q)|)内判定。

注意，关联图的顶点覆盖数的可处理性并不会立即推广到原始图，因此命题 II.5和定理 III.1都不是定理 V.1的直接推论；事实上，原始图较小的顶点覆盖数仍然允许存在任意数量的不同子句。

现在，我们准备给出关于关联图反馈边数的算法。有趣的是，与顶点覆盖数不同，即使对于原始图，反馈边数下CQSAT的参数化复杂性也一直悬而未决。尽管原始图和关联图的反馈边数再次成为相互正交的参数（例如，考虑两个变量在多个子句中共同出现的情况），我们将证明，关联图的算法基本上可以利用针对原始图所发展的技术得到。因此，我们将首先给出原始图反馈边数的结果，这也构成了本节的主要技术贡献。我们通过证明核化算法的存在性来建立该结果。

定理 V.2（$\star$）。设 Q 是一个合取范式量化布尔公式。在多项式



[1]图 8：量化布尔公式 Q的结构感知归约 $\mathcal{R}$ 示意图。（左）：原始图 G_Q 与深度为 h的树深度分解对齐，其中 S 是 T 中高度为 $\mathcal{O}(h)$ 的一条路径；虚线表示祖先与后代之间可能存在的边。（右）：对应的原始图 $G_{Q'}$（简化后）与树深度分解 T'对齐，并展示了 T'中一条高度为 $\mathcal{O}(\log(h))$ 的对应路径 S'； Q'和 S'由 $\mathcal{R}(Q,S)$得到。

时间内，我们可以构造一个等价的合取范式量化布尔公式，它至多包含$12k-8$个变量和 $10k-9+3^{\lfloor(\sqrt{24k+1}+1)/2\rfloor}$个子句，其中 k 是 G_Q的反馈边数。

核化背后的主要思路如下。给定 Q，我们首先在多项式时间内计算原始图 $G{=}G_Q$ 的最小反馈边集 D 。然后，图$H{=}G{-}D$ 是 G的一棵（生成）森林。我们引入一系列归约规则，以缩小 Q 和 H的规模。首先注意到，我们可以删除单位子句（即只包含一个文字的子句）和纯文字（即在 Q中只以正文字或只以负文字出现的变量）。接着，我们考虑 H中的干净边，即不出现在 G任何三角形中的边。注意，H 中至多有 $2\cdot|D|$条边不是干净边，因为 H 中任何不与 D 中的边相邻的边必然是干净边。关键在于，干净边的端点只能共同出现在大小至多为 2 的子句中。这一性质使我们能够大幅简化公式 Q （使用三条归约规则，其正确性可由 Hintikka 策略 [32]得出），从而可以假设每条干净边的两个端点恰好包含在 Q的一个子句中。这使我们能够引入一条简单的归约规则，删除 H 中不是 D中某条边端点的所有叶节点。于是，在约化实例中， H 至多有 $2\cdot|D|$ 个叶节点，因此度（相邻顶点的数量）大于 2 的顶点至多有 $2\cdot|D|-2$个。我们的最后一条归约规则较为复杂，且基于 Hintikka 策略；它通过证明 H 中任意一条极大的（干净）度为 2 的顶点路径都必须包含一个可以“删除”的变量（最内层变量），从而约化 H 中的度为 2 的顶点。这表明 H 的规模有界，因此 G 的规模也有界，进而使我们能够得到 Q的一个界。

根据定理 V.2，我们知道预处理后可以在核（输出）上进行蛮力搜索。由此立即得到一个用于 CQSAT 的单指数固定参数可处理算法。

推论 V.3。以原始图的反馈边数为参数时，CQSAT 是固定参数可处理的。

有趣的是，如果我们将研究范围限制在 c-合取范式量化布尔公式上，我们的核化甚至可以得到一个线性核，即核的大小仅与参数呈线性关系。

推论 V.4。设 c 为一个整数；设 Q 为一个 c-合取范式量化布尔公式。在多项式时间内，我们可以得到一个等价的 c-合取范式量化布尔公式，其中至多包含 $12k-8$ 个变量和 $10k-9+3k(3^c/\binom{c}{2})$个子句，使得 k 是 G_Q的最小反馈边集的大小。

类似地，我们可以证明，对于关联图的反馈边数，存在一个更小的核。

定理 V.5（$\star$）。设 Q 为一个反馈边数为 k 的 I_Q的合取范式量化布尔公式。在多项式时间内，我们可以构造一个包含 $\leq 24k-17$ 个变量和子句的等价合取范式量化布尔公式。

VI. 原始图上的 CQSAT 可处理性

上文的结果全面描绘了 CQSAT 关于关联图的精细复杂性。然而，在考察原始图时，顶点覆盖数和反馈边数对应的可处理性，与已知的关于树宽的不可解性之间存在差距。为了解决这一问题，可以考察 CQSAT 关于反馈顶点数和树深度这些参数的复杂性。我们正在推进树深度方面问题的解决；该问题不仅完全处于开放状态，而且现有技术甚至无法处理树深度的某些重要限制。这类参数是 c-删除集，即到大小至多为 c 的分量的删除距离；众所周知，它处于顶点覆盖数和树深度之间，参见关于顶点完整性的相关工作 [39]。我们提出了三种新算法，每种算法都代表着将 CQSAT 关于顶点覆盖数的可处理性推广开来的一步。每种算法都采用不同的方法，为树深度问题提供了新的、颇具前景的洞见。

首先，我们证明可以高效计算 c-删除集的不同变体，这一点可由以下命题实现。设 $\mathcal{P}(Q,D,c)$是一个性质，对于 CQBF Q 和 Q的 c-删除集 D ，该性质可以为真或假。若存在一个算法，对于给定的 Q、 D和 c ，能够在以 $|D|+c$为参数的固定参数可处理时间内判定$\mathcal{P}(Q,D,c)$是否成立，则称 $\mathcal{P}$ 是可高效计算的。

命题 VI.1（⋆）。设 $\mathcal{P}$ 是任意可高效计算的性质，且 Q 是一个 CQBF。于是，计算满足 $\mathcal{P}(Q, D, c)$的 Q 的最小 c-删除集 D，是一个以 $|D| + c$为参数的固定参数可处理问题。

我们可以消去 CQBF 的 c-删除集 D 中的所有全称变量，而不破坏公式的结构；也就是说，我们得到的公式不会过大，并且仍有一个大小至多为 $2^c c$ 的 $2^c c$-删除集。

命题 VI.2（⋆）。设 Q 是一个 CQBF，且 D 是 Q 的一个 c-删除集。那么，在时间 $\mathcal{O}(2^u \|Q\|)$ 内，其中 $u = |D \cap \mathrm{var}^\forall(Q)|$，我们可以构造一个等价的 CQBF$Q'$ 和一个集合 $D' \subseteq \mathrm{var}^\exists(Q')$，满足 $|D'| \leq 2^u |D|$ s.t. D' 是 Q' 的一个 $2^u c$-删除集。

A. $\exists \leqslant 1 \forall$ 类型的分量

以 c-删除集的大小为参数时，CQSAT 是固定参数可处理的，其分量形式为 $\exists^{\leq 1}\forall$；也就是说，每个分量至多有一个存在变量在前缀中先于其所有（数量任意的）全称变量出现。

定理 VI.3。将 $k + c$ 作为参数时，CQSAT 是固定参数可处理的，其中 k 是一个最小的c-删除集的大小，该删除集将公式分解为形如 $\exists^{\leq 1}\forall$ 的分量。

这推广了以顶点覆盖为参数的 CQSAT 的固定参数可处理性，因为每个分量可以包含任意多个变量以及一次量词交替，而不是只能包含一个变量。

检查每个分量是否为形如 $\exists^{\leq 1}\forall$ 的形式可以在多项式时间内完成。由于根据命题VI.1，我们可以在以其大小加上 c 为参数的固定参数可处理时间内，计算一个最小的 c-删除集，将公式分解为形如 $\exists^{\leq 1}\forall$ 的分量，因此只需证明以下结论。

定理 VI.4（⋆）。设 Q 是一个 CQBF，且 $D \subseteq \mathrm{var}(Q)$ 是一个针对 Q 的 c-删除集，该删除集将公式分解为形如$\exists^{\leq 1}\forall$的分量。那么，判定 Q 时，以 $|D| + c$ 为参数是固定参数可处理的。

定理 VI.4 的证明的主要要素是引理 XIII.3（⋆）。它允许我们移除每种分量类型中除至多 2^c 个分量之外的所有分量。结合对分量类型数量的界定，我们将 Q 化简为一个大小有界的公式，并对其进行蛮力搜索。

B. 单变量删除集

接下来，我们考虑仅由单个变量 e 构成的删除集，但限制在某个分量中出现的变量的量词前缀可以具有任意形状。不失一般性，我们可以假设 e 是存在量化的，并且位于最内层。

该论证涉及一场评估博弈：全称玩家和存在玩家按照量词前缀的顺序轮流为各自的变量赋值。全称玩家试图进行赋值，使存在玩家无法为 e 赋值以满足所有子句。如果全称玩家有一种策略能够确保存在玩家无法选择某个 e 赋值，我们就说该策略禁止了这个赋值。当且仅当全称玩家能够禁止这两个赋值时，量化布尔公式为假。

由于各分量不共享全称变量，因此全称策略可以分解为在各个分量中进行的策略。如果存在不同的分量，其中全称玩家分别可以禁止赋值 $e \mapsto 0$ 和 $e \mapsto 1$，则可以将相应的策略组合成一个全称获胜策略。当存在一个分量使全称玩家必须选择要禁止的赋值，而存在玩家也必须类似地选择在其余分量中进行的赋值时，会出现一个有趣的情形。当且仅当全称玩家能够选择要禁止的赋值的最晚位置（即最内层变量）晚于存在玩家能够选择要进行的赋值的最晚位置时，全称玩家获胜。对这些直觉进行形式化推导即可得到以下结论。

定理 VI.5（⋆）。设 Q 是具有大小为 1 的 c删除集的合取范式量化布尔公式。判定 Q 是否成立是关于 c参数的固定参数可处理问题。

C. 各类型具有多个分量的公式

设 Q 是一个合取范式量化布尔公式，并设 $D \subseteq \mathrm{var}(Q)$是 Q的一个大小为 c的删除集。此外，设 Q' 和 D' 分别是 D 中使用命题 VI.2 消除所有全称变量后得到的合取范式量化布尔公式，以及 Q' 的大小至多为$2^c|D|$ 的 $2^c c$删除集。若 Q' 的每种分量类型都由至少 $2^{|D'|}$ 个分量组成，则称 D' 是全称完备的。我们证明，CQSAT 关于最小全称完备 c删除集的大小参数化时是固定参数可处理的。

定理 VI.6。CQSAT 关于 $k + c$是固定参数可处理的，其中 k 是最小全称完备 c删除集的大小。

这一结果乍看之下令人惊讶且有悖直觉，因为它似乎表明，判定一个较大的合取范式量化布尔公式 Q（其中每种类型都有许多分量）很简单，而我们尚不清楚这对Q的子公式是否成立。然而，我们证明，同一类型的大量分量使全称玩家能够针对每种类型执行所有可能的局部反制策略。这使得前缀中各个分量的相对顺序变得无关紧要。

注意，根据命题 VI.2，判定给定的 c-删除集是否为全称完备是一种可高效计算的性质。这意味着，我们可以在以其大小加上 c 为参数的固定参数可处理时间内，计算出一个最小的全称完备 c-删除集，这一点由命题 VI.1 得出。因此，为证明定理 VI.6，我们假设给定了一个最小的全称完备 c-删除集 D，其对应于 Q。此外，根据命题 VI.2，我们可以假设给定了相应的 Q' 和 D'。只需证明：

定理 VI.7（⋆）。设 Q 是一个 CQBF，且 $\mathrm{var}^{\exists}(Q)$ 是 Q 的全称完备 c-删除集。那么，判定 Q 是固定参数可处理的，其中参数为 $|D| + c$。

VII. 结论

我们研究了在结构限制下对量化布尔公式（QSAT）的求值。尽管原始图上的经典复杂性以及树宽和顶点覆盖数这两个参数已得到充分理解，我们关注的是关联图以及这两个参数之间的差距。我们给出了新的上界和下界，并为 QSAT 在该图最基本的图结构参数方面建立了全面的复杂性理论图景。由此，我们进一步明确了参数化中可以省略量词深度的参数与不能省略量词深度的参数之间的界限，对关联图参数的情况给出了近乎完整的图景，参见图 1。我们通过设计结构感知（SAW）归约，证明了关于反馈顶点数和树深度的下界。随后，我们通过反馈边数的固定参数可处理（fpt）结果，补充了关于顶点覆盖数的已有上界。

尽管本文弥合了许多空白，并加深了我们对 QSAT 在结构参数下困难性的认识，但对于原始图参数下的 QSAT，问题尚未完全解决。一个子句就足以造成差别：如果省略由一个子句引出的边，我们关于反馈顶点数和树深度的下界仍然成立，如图 2（左）所示。为了填补这一空白并分析原始图的树深度，我们首先证明了删除集参数若干变体的固定参数可处理性。

技术：尽管结构感知（SAW）归约背后的思想已在有限的情境中被隐式运用，例如 [26], ，但我们将这一技术形式化并加以全面发展。对于各种参数，当我们关注指数时间假设（ETH）下的精确下界时，我们建立了一个模板，用于设计从 QSAT 到 QSAT 的特定自归约。如图 2（右）所示，SAW 归约使我们能够以一个参数（矩阵结构）的指数级减小为代价，换取运行时间对另一个参数依赖的指数级增长（塔高增加）。

未来工作：我们的分析提出了几个有趣的问题：以原始图的反馈顶点数或树深度为参数时，CNF 上的 QSAT 是否是固定参数可处理的？我们对这两种可能结果都有一些迹象。（N）我们的下界已经非常接近极限，因为仅增加一个子句，就会使树深度和反馈顶点数参数下的问题变得不可解。这指向了困难性，而我们的 SAW 归约或许能为理解并获得原始图上针对其中一个或两个参数的不可解性提供一个良好的起点。（Y）我们针对{v1}删除集的变体所开发的算法技术则指向了另一个方向，并可能成为获得固定参数可处理结果的基础。虽然我们关于关联图反馈顶点数的下界在指数时间假设下是紧的，但对于树深度而言，这一点尚未解决。此外，在强指数时间假设下的下界可能也很有趣。另一个问题是，我们的技术是否能够推广到 DQBF [4] 和 QCSP [25]。

最后，我们预计 SAW 归约将成为解决多项式层次中问题的有用工具。事实上，许多具有实际意义的问题都将受益于精确界；例如，参见 [46], [26]。其中一些问题在其他领域也高度相关，例如可解释性 [14]。

参考文献

[1] C. Ans´otegui、C. P. Gomes 和 B. Selman，《量化布尔公式的致命弱点》，载于 AAAI'05。AAAI出版社，2005年。[2] B. Aspvall、M. F. Plass 和 R. E. Tarjan，《检验某些量化布尔公式真值的线性时间算法》，《信息处理快报》，1979年。[3] A. Atserias 和 S. Oliva，《有界宽度量化布尔公式是多项式空间完备的》，《计算机与系统科学杂志》，2014年。[4] S. Azhar、G. Peterson 和 J. Reif，《不完全信息下多人非合作博弈的下界》，《计算机与数学及其应用杂志》，2001年。[5] P. Beame、R. Impagliazzo、T. Pitassi 和 N. Segerlind，《DPLL中的公式缓存》，《ACM计算逻辑汇刊》，2010年。[6] M. Benedetti 和 H. Mangassarian，《基于量化布尔公式的形式化验-证：经验与展望》，《可满足性、布尔建模与计算杂志》，2008年。[7] A. Biere、M. Heule、H. van Maaren 和 T. Walsh 编，《可满足性手册》（第2版）。荷兰阿姆斯特丹：IOS Press，2021年。[8] R. Bloem、U. Egly、P. Klampfl、R. K¨onighofer 和 F. Lonsing，《基于SAT的电路综合方法》，载于 FMCAD'14。IEEE，2014年。[9] J. A. Bondy 和 U. S. R. Murty，《图论》，数学研究生教材丛书。施普林格，2008年。[10] F. Capelli 和 S. Mengel，《通过知识编译实现可解的量化布尔公式》，载于 STACS'19。达格施图尔出版社，2019年。[11] T. F. N. Chan、J. W. Cooper、M. Kouteck´y、D. Kr´al 和 K. Pek´arkov´a，《最优树深度矩阵与一种用于整数规划的行不变参数化算法》，《计算杂志》，2022年。[12] H. Chen，《量化约束满足与有界树宽》，载于 ECAI'04，2004年。[13] M. Cygan、F. V. Fomin、Ł. Kowalik、D. Lokshtanov、D. Marx、M. Pilipczuk、M. Pilipczuk 和 S. Saurabh，《参数化算法》。施普林格，2015年。[14] A. Darwiche，《逻辑在人工智能中的三种现代作用》，载于 PODS'20。ACM，2020年。[15] R. de Haan、I. A. Kanj 和 S. Szeider，《CSP的亚指数时间复杂性》，《人工智能研究期刊》，2015年。[16] H. de Rider，《图类及其包含关系信息系统（ISGCI）》，2001年至2014年。[17] R. Dechter，《约束满足问题的可解结构》，载于《约束规划手册》。Elsevier，2006年。[18] R. Dechter 和 J. Pearl，《约束网络的树聚类》，《人工智能》，1989年。[19] R. Diestel，《图论》，第4版，数学研究生教材丛书。施普林格，2012年。[20] R. G. Downey 和 M. R. Fellows，《参数化复杂性基础》，计算机科学教材丛书。施普林格，2013年。[21] P. G. Drange、M. S. Dregi 和 P. van 't Hof，《顶点完整性与分量阶连通性的计算复杂性》，《Algorithmica》，2016年。

[22] U. Egly、T. Eiter、H. Tompits 和 S. Woltran，《使用量化布尔公式解决高级推理任务》，载于 AAAI’00，2000年。[23] E. Eiben、R. Ganian、D. Knop、S. Ordyniak、M. Pilipczuk 和 M. Wrochna，《整数规划与关联树深度》，载于 IPCO’19。施普林格，2019年。[24] J. Fichte、M. Hecher、Y. Mahmood 和 A. Meier，《面向论证和树宽的分解引导归约》，载于 IJCAI’21。ijcai.org，2021年。[25] J. K. Fichte、M. Hecher 和 M. F. I. Kieler，《面向树宽的量词消去与展开：QCSP》，载于 CP’20。施普林格，2020年。[26] J. K. Fichte、M. Hecher 和 A. Pfandler，《有界树宽量化布尔公式的下界》，载于 LICS ’20：第35届ACM/IEEE计算机科学逻辑年会。ACM，2020年。[27] J. Flum 和 M. Grohe，《参数化复杂性理论》，《理论计算机科学》丛书。施普林格，2006年。[28] E. C. Freuder，《回溯有界搜索的充分条件》，《ACM杂志》，1985年。[29] R. Ganian 和 S. Ordyniak，《ILP分解参数的复杂性图景》，《人工智能》，2018年。[30] I. P. Gent、P. Nightingale、A. Rowley 和 K. Stergiou，《求解量化约束满足问题》，《人工智能》，2008年。[31] A. Goultiaeva、M. Seidl 和 A. Biere，《弥合双重传播与基于合取范式的量化布尔公式求解之间的差距》，载于 DATE’13。ACM，2013年。[32] E. Gr¨adel、P. G. Kolaitis、L. Libkin、M. Marx、J. Spencer、M. Y. Vardi、Y. Venema 和 S. Weinstein，《有限模型理论及其应用》。施普林格，2005年。[33] M. Grohe，《数据库查询的参数化复杂性》，载于 PODS’01。ACM，2001年。[34] R. Impagliazzo、R. Paturi 和 F. Zane，《哪些问题具有强指数复杂性? 》，《计算机与系统科学杂志》，2001年。[35] C. Jordan、W. Klieber 和 M. Seidl，《使用QCIR进行非合取范式量化布尔公式求解》，载于 AAAI’16。AAAI出版社，2016年。[36] H. Kleine B¨uning 和 T. Lettman，《命题逻辑：演绎与算法》。纽约：剑桥大学出版社，1999年。[37] T. Korhonen，《用于树宽的单指数时间2-近似算法》，载于 FOCS’22，2022年。[38] M. Lampis 和 V. Mitsou，《重新审视带量词交替的树宽》，载于 IPEC’17。达格施图尔出版社，2017年。[39] ——，《顶点完整性的精细化元定理》，载于 ISAAC’21。LIPIcs，2021年。[40] C. Lautemann，《有界错误概率多项式时间与多项式层次》，《信息处理快报》，1983年。[41] J. Li 和 J. Nederlof，《在 O*(2.7k) 时间内检测大小为 k 的反馈顶点集》，载于 SODA’20。工业与应用数学学会（SIAM），2020年。[42] D. Marx 和 V. Mitsou，《以树宽为参数的可选择性问题的双指数和三指数界》，载于 ICALP’16。达格施图尔出版社，2016年。[43] J. Ne ̆set ̆ril 和 P. O. de Mendez，《稀疏性：图、结构与算法》。施普林格，2012年。[44] R. Niedermeier，《固定参数算法导论》，牛津数学及其应用讲座丛书。美国纽约州纽约市：牛津大学出版社，2006年。[45] C. Otwell、A. Remshagen 和 K. Truemper，《一种用于规划问题的有效量化布尔公式求解器》，载于 AMCS’04。CSREA出版社，2004年。[46] G. Pan 和 M. Y. Vardi，《多项式空间中的固定参数层次》，载于 LICS。IEEE计算机学会，2006年。[47] C. H. Papadimitriou，《计算复杂性》。Addison-Wesley，1994年。[48] M. Pilipczuk 和 M. Sorge，《不同向量问题的双指数下界》，《离散数学与理论计算机科学》，2020年。[49] F. Reidl、P. Rossmanith、F. S. Villaamil 和 S. Sikdar，《一种更快的树深度参数化算法》，载于 ICALP’14。施普林格，2014年。[50] J. Rintanen，《通过定理证明器构造条件计划》，《人工智能研究期刊》，1999年。[51] N. Robertson 和 P. D. Seymour，《图小子 I：排除一个森林》，《组合理论杂志B辑》，1983年。[52] ——，《图小子 X：树分解的障碍》，《组合理论杂志B辑》，1991年。[53] M. Samer 和 S. Szeider，《命题模型计数算法》，《离散算法杂志》，2010年。[54] ——，《重新审视有界树宽的约束满足》，《计算机与系统科学杂志》，2010年。[55] ——，《固定参数可处理》，载于《可满足性手册》第二版，《人工智能与应用前沿》系列。IOS Press，2021年。[56] A. Shukla、A. Biere、L. Pulina 和 M. Seidl，《量化布尔公式应用综述》，载于 ICTAI’19。IEEE，2019年。[57] F. Slivovsky 和 S. Szeider，《一种按关联树宽参数化的更快命题模型计数算法》，载于 SAT’20。施普林格，2020年。[58] M. Sorge 和 M. Weller，《图参数层级》，2012–2020年。[59] L. J. Stockmeyer 和 A. R. Meyer，《需要指数时间的字问题》，载于 STOC’73。ACM，1973年。[60] M. M. Syslo，《外平面图的刻画》，《离散数学》，1979年。[61] S. Toda，《PP 与多项式时间层级一样困难》，《SIAM 计算杂志》，1991年。[62] G. Tseytin，《关于命题演算中推导的复杂性》，《构造性数学与数理逻辑研究》，第二部分，《数学研讨会》，1970年。[63] V. V. Vazirani，《近似算法》。施普林格，2003年。[64] R. Wallon 和 S. Mengel，《重新审视合取范式编码的图宽度测度》，《人工智能研究期刊》，2020年。[65] R. Williams，《量化布尔公式算法》，载于 SODA’02。ACM，2002年。

附录

VIII. 补充预备知识

为便于介绍算法，我们等价地以另一种（原子）形式查看量化布尔公式 $Q = Q_1 v_1 Q_2 v_2 \cdots Q_n v_n . F$，其中 v_j 中的变量是 $1 \le j \le n$ 的变量，并且对于 $2 \le i \le n$ 我们没有 $Q_i \neq Q_{i-1}$。

a) 欣蒂卡博弈：设 $Q = Q_1 v_1 Q_2 v_2 \cdots Q_n v_n . F$ 是一个量化布尔公式。设 V 是一个变量集合。我们用 $V^{\exists}$ 和 $V^{\forall}$ 分别表示 V 中所有存在玩家变量和全称玩家变量的集合。对于变量 $v \in \mathrm{var}(Q)$，我们用 $V_{<v}$ ($V_{>v}$) 表示 V 中在 v 之前（之后）出现在 Q 前缀中的所有变量的集合。类似地，对于赋值 δ：$V' \to \{0,1\}$，其中 $V' \subseteq \mathrm{var}(Q)$，我们用 $\delta_{<v}$ ($\delta_{>v}$) 表示将赋值 δ 限制在 $V_{<v}$ ($V_{>v}$) 中所有变量上的结果；用 δ^Q 表示对于 $Q \in \{\exists, \forall\}$，将赋值 δ 限制在 V^Q 中所有变量上的结果。最后，对于定义在 $\mathrm{var}(Q)$ 中两个互不相交的变量集上的赋值 δ 和 δ'，我们用 $\delta \cup \delta'$ 表示这两个赋值的不相交组合，即，$\delta \cup \delta'$ 等于通过如下方式定义的赋值 δ''：$\mathrm{var}(\delta) \cup \mathrm{var}(\delta') \to \{0,1\}$：若 $v \in \mathrm{var}(\delta)$，则令 $\delta''(v) = \delta(v)$；若 $v \in \mathrm{var}(\delta')$，则令 $\delta''(v) = \delta'(v)$。

为便于表示，以下令 $V = \mathrm{var}(Q)$。Eloise 的策略（存在策略）是一列映射 $\mathcal{T} = (\tau_v : \{0,1\}^{V^{\forall}_{<v}} \to \{0,1\})_{v \in V^{\exists}}$。对于 $\delta : V^{\forall} \to \{0,1\}$，令 $\alpha(\mathcal{T}, \delta) : \mathrm{var}(Q) \to \{0,1\}$ 表示通过如下设定所定义的赋值：

- $\alpha(\mathcal{T}, \delta)(y) = \delta(y)$ if $y \in V^{\forall}$ and
- $\alpha(\mathcal{T}, \delta)(x) = \tau_x(\delta_{<x})$ if $x \in V^{\exists}$.

若对于任意赋值 $\delta : V^{\forall} \to \{0,1\}$（以下也称为全称方博弈），公式 F 在赋值 $\alpha(\mathcal{T}, \delta)$ 下都成立，则存在策略 $\mathcal{T}$ 称为获胜策略。

Abelard 的策略（全称策略）定义方式类似，只需交换映射 δ 和 τ；若 F 不成立，则称全称策略为获胜策略。形式上，它是一列映射 $\Lambda = (\lambda_v : \{0,1\}^{V^{\exists}_{<v}} \to \{0,1\})_{v \in V^{\forall}}$。若对于任意赋值 $\delta : V^{\exists} \to \{0,1\}$（以下也称为存在性博弈），公式 F 在按存在玩家情形类似定义的赋值 $\alpha(\Lambda, \delta)$ 下为假，则全称策略 Λ 称为获胜策略。也就是说，$\alpha(\Lambda, \delta) : V \to \{0,1\}$ 表示通过如下设定所定义的赋值：

- $\alpha(\Lambda, \delta)(x) = \delta(x)$ 当且仅当 $x \in V^{\exists}$ 且
- $\alpha(\Lambda, \delta)(y) = \lambda_y(\delta_{<y})$ 当且仅当 $y \in V^{\forall}$。

设 $\mathcal{T} = (\tau_v : \{0,1\}^{V^{\forall}_{<v}} \to \{0,1\})_{v \in V^{\exists}}$ 是一个存在玩家策略，并设 $\Lambda = (\lambda_v : \{0,1\}^{V^{\exists}_{<v}} \to \{0,1\}_{v \in V^{\forall}}$ 是一个全称玩家策略。随后，以 $\mathcal{T}$ 对抗 Λ 进行对局会产生一个赋值 $\alpha(\mathcal{T}, \Lambda)$：$V \to \{0,1\}$ of$Q$，其可以递归定义如下。设 α_0 为空赋值。此外，对于每个满足 $1 \le i \le n$ 的 i，我们区分两种情况。如果 $Q_i = \exists$，那么 α_i 是通过赋值 $\tau_{v_i}(\alpha^{\forall}_{i-1})$ 对 α_{i-1} 进行扩展而得到的，其中 v_i。类似地，如果 $Q_i = \forall$，那么 α_i 是通过赋值 $\lambda_{v_i}(\alpha^{\exists}_{i-1})$ 对 α_{i-1} 进行扩展而得到的，其中 v_i。最后，我们令 $\alpha(\mathcal{T}, \Lambda) = \alpha_n$。注意，$\alpha(\mathcal{T}, \Lambda)$ 会产生存在性博弈 $\alpha(\mathcal{T}, \Lambda)^{\exists}$ 和全称方博弈 $\alpha(\mathcal{T}, \Lambda)^{\forall}$，并且有 $\alpha(\mathcal{T}, \Lambda) = \alpha(\mathcal{T}, \alpha(\mathcal{T}, \Lambda)^{\forall})$，以及 $\alpha(\mathcal{T}, \Lambda) = \alpha(\Lambda, \alpha(\mathcal{T}, \Lambda)^{\exists})$。

从 $V^{\forall}$ 的某个子集到 $\{0,1\}$ 的一个映射 δ 称为全称方博弈；类似地，从 $V^{\exists}$ 的某个子集到 $\{0,1\}$ 的一个映射 δ 称为存在性博弈。

命题 VIII.1（众所周知）。量化布尔公式 Q 为真，当且仅当在 Q 上存在获胜的存在策略；这又当且仅当在 Q 上不存在获胜的全称策略。

b) 核化：核化是一种算法，对于给定的实例 $(\mathcal{I}, k) \in \Sigma^* \times \mathbb{N}$，在时间 $\mathcal{O}(\mathrm{poly}(\|\mathcal{I}'\| + k))$ 内输出一对 $(\mathcal{I}', k') \in \Sigma^* \times \mathbb{N}$，满足：(i) $(\mathcal{I}, k) \in L$ 当且仅当 $(\mathcal{I}', k') \in L$；以及 (ii) $\|\mathcal{I}\| + k' \le g(k)$，其中 g 是任意可计算函数，称为核的大小。如果 g 是一个多项式，则称 L 具有多项式核。众所周知，一个参数化问题是固定参数可处理的，当且仅当它是可判定的并且具有核化 [20]。

c) 反馈边集：对于这一参数，我们使用以下结果。

命题 VIII.2（众所周知）。设 G 为一个无向图。那么，可以在多项式时间内为 G 计算出一个最小反馈边集。

证明。通过为 G 计算一个生成森林，并将生成森林之外的所有边取入 D，即可在多项式时间内计算出一个最小反馈边集 D。□

IX. 第 III 节省略证明

定理 III.1。存在一种算法，对于具有 G_Q 顶点覆盖数 k 的合取析取范式量化布尔公式 Q，可在 $2^{2^{\mathcal{O}(k)}} \cdot \mathrm{poly}(\|Q\|)$ 时间内判定 Q 是否为真。如果 Q 属于 d-合取析取范式，该算法的运行时间为 $2^{k^{\mathcal{O}(d)}} \cdot \mathrm{poly}(\|Q\|)$。

证明。设 Q 是一个量化布尔公式，其矩阵为 $F = C \wedge D$，其中 C 为合取范式，D 为析取范式。进一步，设 X 是原始图的一个顶点覆盖，满足 $|X| = k$，且可在 $2^k \cdot poly(\|Q\|)$ 时间内计算得到。为简单起见，我们假设每个量词块仅包含一个变量，因此 Q 的前缀可以写为 $Q_1 v_1 \ldots Q_n v_n$。考虑一种回溯算法，该算法维护一个变量赋值 α：对于某个 $1 \le i \le n$，其形式为 $\{v_1, \ldots, v_i\} \to \{0,1\}$；同时维护一个缓存，其中包含若干对 (F', val)，这些二元组由之前遇到的矩阵子公式 F' 及其真值 $\mathrm{val} \in \{0,1\}$ 组成。算法首先检查缓存中是否存在条目 $(F[\alpha], \mathrm{val})$；如果存在，则返回相应的值。如果缓存中不存在该条目，则递归地按如下方式确定 $Q_{i+1} v_{i+1} \ldots Q_n v_n . F[\alpha]$ 的真值 val：如果

$i = n$，不存在未赋值变量，并且当 F 被 α 满足时，val $= 1$；当 F 被否定时，val $= 0$。否则，算法在下一个未赋值变量 v_{i+1} 上进行分支，并分别使用赋值 $\alpha \cup \{v_{i+1} \mapsto 0\}$ 和 $\alpha \cup \{v_{i+1} \mapsto 1\}$ 递归调用自身。如果 v_{i+1} 为存在量化，算法将 val 设为返回值中的最大值；如果为全称量化，则将其设为最小值。随后，算法将二元组 $(F[\alpha]$, val) 添加到缓存中，并返回 val。

显然，该算法是正确的。忽略递归调用，它的运行时间为 $poly(K)$，其中 K 是缓存中的条目数。搜索树中的节点数和 K 都由对于部分赋值 $\alpha : \{v_1, \ldots, v_i\} \to \{0,1\}$的子公式数量 $F[\alpha]$ 的数目所界定。我们论证，对于每个 $1 \le i \le n$，此类子公式至多有 $2^k \cdot 2^{(3^k)^2}$ 个。一个部分赋值 α 可以写成不相交并集 $\alpha = \alpha_X \cup \beta$，其中 α_X 为顶点覆盖 X中的变量赋值，而 β 为顶点覆盖之外的变量赋值。我们还可以将 F 的子句和项划分为$F = F' \cup F_{>i}$，其中 $F_{>i}$ 不包含 $\{v_1, \ldots, v_i\}$中的变量，因此 $F[\alpha] = F'[\alpha] \cup F_{>i}$。只需界定不同子公式$F'[\alpha]$的数量即可。进一步地，我们将 F'划分为 $F' = F_X \cup F''$，其中 var(F_X) $\cap \{v_1, \ldots, v_i\} \subseteq X$，且$F''$中的每个子句或项都包含 $\{v_1, \ldots, v_i\} \setminus X$中的一个变量。这使我们可以写成 $F'[\alpha] = F'[\alpha_X][\beta] = F_X[\alpha_X] \cup F''[\alpha_X][\beta]$。子公式 $F_X[\alpha_X]$ 和$F''[\alpha_X]$至多有 2^k 个。对于每个子公式 $F''[\alpha_X]$，子公式$F''[\alpha_X][\beta]$通过给不出现在顶点覆盖 X 中的变量赋值而得到，这意味着它们的所有邻居都必须位于顶点覆盖内。因此，$F''[\alpha_X][\beta]$是由 X中的变量构造的子公式。在 k 个变量上至多有 3^k 个子句（或项），因此至多有 2^{3^k} 个 CNF（或 DNF）公式。由于 F 由一个 CNF 和一个 DNF 组成，因此至多有 $2^{(3^k)^2}$ 个 C DNF 公式，而总体上至多有 $2^k \cdot 2^{(3^k)^2}$ 个子公式 $F'[\alpha]$，这属于 $2^{3^{\mathcal{O}(k)}}$。对于 d-CDNF，大小至多为 d的子句或项有 $(2k+1)^d$ 个，因此子公式数量的上界为 $2^{k^{\mathcal{O}(d)}}$。由于对于每个 $1 \le i \le n$，我们得到 α 的不同定义域，因此缓存条目数和搜索树中的节点数有 $n \cdot 2^{3^{\mathcal{O}(k)}}$ 的上界，或者在 d-CDNF 中为 $n \cdot 2^{k^{\mathcal{O}(d)}}$。因此，对于一般的 CDNF，总体运行时间为 $2^{2^{\mathcal{O}(k)}} \cdot \text{poly}(\|Q\|)$；对于 d-CDNF，则为 $2^{k^{\mathcal{O}(d)}} \cdot \text{poly}(\|Q\|)$。

□

定理 III.2（QSAT$_\ell$ 与树宽的上界）。给定任意量词深度为 ℓ、满足 $k =$tw(G_Q) 的合取析取范式量化布尔公式 Q。则可在时间 tow$(\ell, \mathcal{O}(k)) \cdot \text{poly}(|\text{var}(Q)|)$ 内判定 Q 上的 QSAT$_\ell$。

证明。我们以 $Q = Q_\ell = \exists$为最内层量词的情形来说明证明过程。设 $Q = Q_1 V_1 . \cdots Q_\ell V_\ell . C \wedge D$，并令 $\mathcal{T} = (T, \chi)$ 为一个树分解其 G_Q 宽度为 $\mathcal{O}(k)$，可在 $2^{\mathcal{O}(k)} \cdot \text{poly}(|\text{var}(Q)|)$ [37] 时间内计算得到。对于 T 中的每个节点 t，令其子节点集合为 cld(t)；不失一般性地假设 $|\text{cld}(t)| \le 2$（可通过添加辅助节点获得）。我们使用辅助变量 $S = \{sat_t \mid : \text{在 } T\}$中。然后，我们从 Q 和 $\mathcal{T}$ 构造一个线性 $Q' = Q_1 V_1 . \cdots Q_\ell V_\ell \cup S . C \cup C''$ SAW 归约，定义一个量化布尔公式：（，其矩阵为合取范式：

$$sat_t \to \bigvee_{t' \in \text{cld}(T)} sat_{t'} \vee \bigvee_{d \in D, \text{var}(d) \subseteq \chi(t)} d \quad \text{for every } t \text{ of } T \tag{10}$$

$$sat_r \quad \text{for root } r \text{ of } T \tag{11}$$

公式 (10) 定义了节点 t 上一个项何时满足；结合公式 (11)，可轻松将其转换为合取范式集合 C'（使用分配律）。注意，根节点处必须至少有一个项满足。该归约得到一个针对 $G_{Q'}$ 的树分解（TD） $\mathcal{T}' :=$ (T, χ')，其中对 T 中的每个 t，均有 $\chi'(t) :=$ $\chi(t)$ $\cup \{sat_t\} \cup \{sat_{t'} \mid t' \in \text{cld}(t)\}$。 $\mathcal{T}'$ 的宽度为 width$(\mathcal{T})$ $+ 3 \in \mathcal{O}$(width$(\mathcal{T})$)，因此在 Q' 上，命题 II.3 中的算法运行时间为 tow$(\ell, \mathcal{O}(k)) \cdot \text{poly}(|\text{var}(Q)|)$。□

X. 更受限参数的下界

第 IV-A 节中提出的归约 $\mathcal{R}$ 可以证明能够归约那些比稀疏反馈顶点数限制更严格的参数（引理 IV.1）。

推论 X.1（缩小到稀疏半梯的距离）。给定一个 3,1-CDNF 量化布尔公式 Q，其中 S 是到 Q的稀疏半梯的距离集。那么，归约 $\mathcal{R}$ 构造出一个量化布尔公式 Q'，其到 Q' 的稀疏半梯的距离集为 S'，且 $|S'|$ 有界于 $\mathcal{O}(\log(|S|))$。

证明。该结果由引理 IV.1 以及以下观察得出：只要 S 是到 Q的稀疏半梯的距离集， S' 就是到 Q' 的稀疏毛毛虫的距离集。这个毛毛虫源于 Q 中仅使用变量 v（来自 var(matr$(Q)) \setminus S$）的 3-CNF 子句 c_i，参见图 7 中的 (ii) 类子句。然而，通过为每个这样的 c_i引入一个额外的最内层存在量词副本变量 v_i（对应于 v），可以将该毛毛虫转变为半梯。然后，对于这些副本的任意全序，我们通过 $(v_i \wedge \neg v)$ 和 $(v \wedge \neg v_i)$ 确保 v 与第一个副本 v_i 等价，并依此类推，确保第一个副本与第二个副本等价。我们对公式 (3) 作相应调整：不再让 v在 $G_{Q'}$中拥有所有邻居 sat_i，而是让每个副本 v_i 拥有一个邻居 sat_i，从而将星形子图拆解为半梯。□

推论 X.2（距离稀疏半梯的下界）。给定任意一个量词深度为 ℓ 的 CDNF 中的 QBF Q，以及一个到稀疏半梯 Q 的最小距离集 S，其中 $k = |S|$。那么，在指数时间假设成立的条件下，无法在时间 tow$(\ell, o(k)) \cdot \text{poly}(|\text{var}(Q)|)$ 内判定 QSAT$_\ell$ on Q。

证明。该结果由定理 IV.5 得出，其中使用推论 X.1 代替引理 IV.1。□

有趣的是，要找到相应的距离集 S，只需考虑最内层量词中的变量；检查归约 $\mathcal{R}$和引理 IV.1 后，这一点便显而易见。

推论 X.3（到稀疏半梯的距离的下界（最内层量词））。给定任意一个 CNDF 量化布尔公式 $Q = Q_1V_1 \cdots Q_\ell V_\ell.F$，以及其到稀疏半梯的最小距离集 $S \subseteq V_\ell$，其中 Q 满足 $k = |S|$。那么，在指数时间假设成立的条件下，关于 Q 的 QSAT$_\ell$ 无法在 tow$(\ell, o(k)) \cdot$ poly(|var(Q)|) 时间内判定。

证明。观察到 $\mathcal{R}$ 仅在最内层量词上构造所得距离集 S'，即可由推论 X.2 得出该结果，参见引理 IV.1。□

显然，定理 IV.5 以及推论 X.2、X.3 的结果可立即分别推广到（非稀疏的）反馈顶点数和到半梯的距离，参见图 6。不过，事实证明，这一结果还可以进一步加强，如下文所示。具体而言，归约 $\mathcal{R}$ 对一种略弱于到毛毛虫图的距离的参数也适用：如果毛毛虫图的高度（最大分量）由某个固定值 h 界定，那么 $\mathcal{R}$ 不会显著增加高度，即所得图的毛毛虫图高度为 $\mathcal{O}(h)$。

推论 X.4. 给定一个任意的、量词深度为 ℓ 的合取析取范式中的量化布尔公式 Q，以及一个到高度有界的、大小为 Q 的稀疏毛毛虫图 $\mathcal{O}(h)$ 的最小距离集 S，其中 $k = |S|$。那么，在指数时间假设下，不能在时间 tow$(\ell, o(k)) \cdot$ poly(|var(Q)|) 内判定在 Q 上的 QSAT$_\ell$。

证明。该结果由推论 X.2 得出，注意到通过 $\mathcal{R}(Q, S)$ 获得的、针对 Q' 的毛毛虫图 $G_{Q'}$ 的高度由 $2 \cdot h + 1$ 有界，即为 $\mathcal{O}(h)$，参见图 7。□

注意，通过采用与推论 X.1 的证明中类似的过程，我们无法将这一结果推广到到半梯子的距离，因为这会将高度增加到 $\mathcal{O}(h \cdot \log(k))$。

有趣的是，当半梯大小（分量数量）有界时，问题 QSAT$_\ell$ 仍然是困难的。

推论 X.5. 给定任意一个量词深度为 ℓ、属于合取析取范式的量化布尔公式 Q，以及一个大小为 m 的稀疏半梯（Q 的）的最小距离集 S，满足 $k = |S|$。那么，在指数时间假设成立的条件下，QSAT$_\ell$ 在 Q 上无法在 tow$(\ell, o(k)) \cdot$ poly(|var(Q)|) 时间内判定。

证明。该结果源于以下观察：如上所述，$\mathcal{R}$可以通过为仅使用 S上的变量的子句添加路径来扩展。这些子句可以很容易地连接到一条（较大的）路径上。因此，对于满足 $c_i \in C$ with var$(c) \subseteq S$ 的子句（“类型 (i) 子句”，参见图 7），公式 (4) 会被修改为：将 sat_i 的每次出现替换为一个由 $sat_j \wedge sat_i$ 构成的合取，其中 j 是满足类型 (i) 子句 c_j 在全序中直接排在 C 的 c_i 之前的最大值。类似地，S 中的元素也可以通过例如添加辅助变量连接到一条路径上。由此得到的、经过修改的 $\mathcal{R}$ 归约记为 $\mathcal{R}'$。然后，可以很容易地推广推论 X.2 以证明该结论，因为对于通过 $\mathcal{R}'(Q, S)$ 得到的 Q'，其半梯 $G_{Q'}$ 的大小有 $m + 2$ 的界。□

然而，如果高度和大小都受到限制，我们可以得到一个类似于顶点覆盖数的结果。

推论 X.6. 给定任意一个量词深度为 ℓ、属于 3,1-CDNF 的量化布尔公式 Q，以及一个大小为 m、高度为 h 的毛毛虫图（G_Q 的）的距离集 S，满足 $k = |S|$。那么，存在一个用于在 Q 上求解 QSAT$_\ell$ 的算法，其运行时间为 $2^{\mathcal{O}((k+hm)^3)} \cdot$poly(|var$(Q)$|)。

证明. 该结果由命题 II.5 得出，因为可以轻松构造一个大小为 $\mathcal{O}(k + h \cdot m)$ 的 G_Q 的顶点覆盖，其中包含 G_Q 的所有顶点。□

XI. 第 IV 节省略的细节与证明

A. 反馈顶点数的紧下界

例 XI.1. 回顾例 II.2 中的量化布尔公式 Q'，其中 matr$(Q') = C \wedge D$。考虑 Q'的稀疏反馈顶点集 $S := \{a, c\}$。由于 $|S| = 2$，我们只需要两个索引，并假设对于每个子句 $c_i \in C$，$a =$ var(lit(c_i, 1)) 始终是第一个变量，而 $c =$ var(lit(c_i, 2)) 是第二个变量。我们进一步假设 $[\![a]\!]_1 := \{\neg idx_1^1\}$、$[\![c]\!]_2 := \{\neg idx_2^1\}$ 以及 $[\![a]\!]_2 := \{idx_2^1\}$、$[\![c]\!]_1 := \{idx_2^1\}$。不过请注意，由于 a (c) 始终分别是第一个（第二个）变量，$[\![a]\!]_2$和 $[\![c]\!]_1$ 不会被使用。 $\mathcal{R}(Q', S)$ 等价于 $Q'' = \forall a, b \exists c, d. \forall idx_1^1, idx_2^1, val_1, val_2, sat, sat_1, sat_2, sat_3, sat_4 C' \vee D'$，下面我们将讨论 C' 对应的 DNF 项以及 D' 的单例子句。

(1) $a \wedge \neg idx_1^1 \wedge \neg val_1$, $\neg c \wedge \neg idx_2^1 \wedge \neg val_2$(2) $\neg a \wedge \neg idx_1^1 \wedge \neg val_1$, $\neg c \wedge \neg idx_2^1 \wedge \neg val_2$(3) $sat_1 \wedge \neg b$, $sat_2 \wedge b$, $sat_3 \wedge d$, $sat_4 \wedge \neg d$(4) $sat_1 \wedge idx_1^1$, $sat_2 \wedge idx_1^1$, $sat_3 \wedge idx_1^1$, $sat_4 \wedge idx_1^1$,$sat_1 \wedge idx_2^1$, $sat_2 \wedge idx_2^1$, $sat_3 \wedge idx_2^1$, $sat_4 \wedge idx_2^1$(5) $sat_2 \wedge val_1$, $sat_4 \wedge val_1$, $sat_1 \wedge val_2$, $sat_2 \wedge val_2$(6) $sat_1 \wedge \neg val_1$, $sat_3 \wedge \neg val_1$, $sat_3 \wedge \neg val_2$, $sat_4 \wedge$ (7) $\neg val_2$ $sat \wedge b$, $sat \wedge \neg d$(8) $\neg sat_1$, $\neg sat_2$, $\neg sat_3$, $\neg sat_4$(9) $\neg sat$ ■

引理 IV.1（降低反馈顶点数）。给定属于 3,1-CDNF 的量化布尔公式 Q 和一个稀疏反馈顶点

集 S，由 Q、$\mathcal{R}$ 构造量化布尔公式 Q'，其稀疏反馈顶点集为 S'，且 Q' 使得 $|S'|$ 属于 $\mathcal{O}(\log(|S|))$。

证明。如上所述，$\mathcal{R}$ 产生了一个稀疏反馈顶点集 S'，其对应的图为 Q'。事实上，(i) $G_{Q'} - S'$得到一个无环图，因为每个 $x \in S$ 在 $G_{Q'} - S'$ 中都是孤立的，而 $G_{Q'} - S'$ 中剩余的边只涉及 sat_i 和 y 的某些 $G_{Q'} - S'$。为了导出矛盾，假设 $G_{Q'} - S$ 包含一个环 x_1、sat_{i_1}、x_2、sat_{i_2}、.....、$sat_{i_{r-1}}$、x_r，其中 $x_r = x_1$。变量 x_j 和 x_{j+1} 在 $G_{Q'}$ 中仅当 x_j、$x_{j+1} \in$ var(c_i)时才与 sat_{i_j} 相邻，因此 x_j和 x_{j+1}在 G_Q中相邻。所以，如果 $r > 2$，我们将在 $G_Q - S$中得到一个环，这与 S 是反馈顶点集的假设矛盾。若 $r = 2$，则该环的形式为 x_1、sat_{i_1}、x_2、sat_{i_2}、x_1，并且存在不同的子句 c_{i_1}、c_{i_2}，使得 x_1、$x_2 \in$var(c_{i_1})且 x_1、$x_2 \in$var(c_{i_2})，这与S 是稀疏的假设矛盾。此外，(ii) $G_{Q'} - S'$ 中相邻顶点可能同时出现的唯一项，是公式 (3) 中的项，因为其他析取范式项至多使用一个不在 S'中的变量。这证明了 S' 是 G_Q的一个稀疏反馈顶点集。最后，根据构造，$|S'| \leq 3 \cdot \lceil \log(|S|) \rceil + 4$，其属于 $\mathcal{O}(\log(|S|))$。 □

定理 IV.2（运行时间）。对于一个 Q in 3,1-CDNF 中的 QBF，其矩阵为 $Q = C \wedge D$，并且变量集合 $S \subseteq$ var(Q) 为 Q 的变量，$\mathcal{R}$ 的运行时间为 $\mathcal{O}(\lceil \log(|S|+1) \rceil \cdot (|S|+|C|)+|D|)$。

证明。共有 $\mathcal{O}(|S|)$ 个公式 (1)、(2) 的实例，每个实例的大小为 $\mathcal{O}(\log(|S|))$。此外，还有 $\mathcal{O}(|C|)$ 个常数大小的公式 (3)、(5) 和 (6) 的实例。最后，还有 $\mathcal{O}(|C|)$ 个公式 (8) 的实例，其大小上界为 $\mathcal{O}(\log(|S|))$，$\mathcal{O}(|C|\log(|S|))$ 个大小为 2 的公式 (4) 的实例，以及 $\mathcal{O}(|D|)$ 个常数大小的公式 (7) 的实例。 □

定理 IV.3（正确性）。给定一个 Q in 3,1-CDNF 中的 QBF，以及变量集合 $S \subseteq$ var(Q) 为 Q 的变量，归约 $\mathcal{R}$ 计算出一个实例 Q'，该实例与 Q 等价。事实上，α 对 matr(Q) 的变量的任意赋值满足 matr(Q)，当且仅当 α 到变量 VarIdxs ∪VarVals ∪的每个扩展 α' 都满足 matr(Q')。

证明。 $\Longrightarrow$：假设 Q 有效。接下来，我们证明，对于 $C \wedge D$的任意满足赋值 α : var(Q) $\rightarrow \{0,1\}$，将 α'扩展到变量 VarIdxs ∪VarVals ∪VarSat 的任意扩展都是 matr(Q') $= C' \vee D'$ 的满足赋值。反设存在这样的扩展 α'，满足 $C'[\alpha'] \neq \{\emptyset\}$ 和 $D'[\alpha'] \neq \emptyset$。那么，Formulas (1)–(7) 给出的 C' 的所有项均不被 α' 满足，并且 Formulas (8) 和（9）给出的 D' 中至少有一个子句也不被 α' 满足。我们区分两种情况。

情况 (a)：公式 (8) 中有些公式不被 α' 满足，即，对于某个 $1 \leq i \leq |C|$，$\alpha'(sat_i) = 1$ 。由于 α' 不满足公式 (4) 中的任何公式，因此第 j 个索引 ($1 \leq j \leq 3$) 精确地指向子句 c_i 的第 j 个变量 v_j : = var(lit(c_i, j))，即 α'(var($[\![v_j]\!]_j$)) = sgn($[\![v_j]\!]_j$)。接着，由于 α' 不满足公式 (1) 和 (2) 中的任何公式，我们知道赋值 α' 按照 α 为 c_i 中位于 S （且满足 $v_j \in S$）的第 j 个变量 v_j 设置值 val_j，即满足 $\alpha'(val_j) = \alpha(v_j)$。然而，由于 α' 不满足公式 (5) 和 (6) 中的任何公式，α 无法通过对任何位于 S 中的变量 y 进行赋值来满足子句 c_i。最后，由于 α 仍满足 c_i，至少存在一个这样的变量 $y \in$ var(c_i) $\setminus S$，满足 $\alpha(y) = \alpha'(y)$，使得 α' 恰好满足公式 (3) 的一个实例，其中 $l \in c_i$ 是关于 y 的文字，即 var(l) $= y$。这与如下假设矛盾：按照公式 (1)–(9) 的构造，α' 既不满足 C'，也不满足 D'。

情况（b）：公式（9）不被 α'满足，即$\alpha'(sat) = 1$。那么，由于 α'不满足 Formulas (7) 中的任何公式，我们可知 α' 不满足任何 $\{l\} \in D$，即对于每个 $\{l\} \in D$，都有 α'(var(l)) $\neq$ sgn(l)。因此，$D[\alpha'] \neq \{\emptyset\}$，故根据 α'的构造，我们有 $D[\alpha] \neq \{\emptyset\}$。然而，这与假设 α 是 matr(Q) 的满足赋值相矛盾。

$\Longleftarrow$：我们通过逆否命题证明这一方向：取任意赋值 α : var(Q) $\rightarrow \{0,1\}$ 不满足 matr(Q)，并证明存在一个将 α' 扩展到变量 VarIdxs ∪VarVals ∪ VarSat 的扩展 α，使得 α' 不满足 matr(Q')。我们再次按情形进行讨论。

情形（a）：$C[\alpha] \neq \emptyset$ 由于至少有一个子句 $c_i \in C$，即 $\{c_i\}[\alpha] \neq \emptyset$。于是，（i）对于任意满足 $i' \neq i$的 $1 \leq i' \leq |C|$，我们令 $\alpha'(sat)$: = 0，$\alpha'(sat_i)$: = 1，并令 $\alpha'(sat_{i'})$: = 0 。此外，对于每个 $1 \leq j \leq 3$，令 v_j : = lit(c_i, j) 为子句 c_i 的第 j 个文字对应的变量。最后，（ii）我们令第 j 个索引指向 v_j，即对于每个 $b \in [\![v_j]\!]_j$ 以及每个 $1 \leq j \leq 3$，设置 α'(var(b)): = sgn(b)；并且（iii）设置第 j 个索引的值，使得 $\alpha'(val_j)$: = $\alpha(v_j)$。因此，根据 α' 的构造，公式（1）或（2）的任何实例都不会被 α' 满足。由于 α 不满足 $\{c_i\}$，可知 α' 不满足公式（3）的任何实例。此外，由于根据 α' 的构造（ii），第 j 个索引指向 v_j，公式（4）中的任一个都无法被 α' 满足。同样地，根据（iii），公式（5）或（6）的任何实例都不会被 α' 满足。最后，根据构造（i），公式（7）都不被 α' 满足，D' 也不被满足，因为例如，并非所有公式（8）都被 α' 满足。

情形（b）：$D[\alpha] \neq \{\emptyset\}$，即对于每个 $\{l\} \in D$，α (l) $\neq$ sgn(l)。在此情形下，(i) 我们令 $\alpha'(sat) := 1$，并且 (ii) 对任意 $1 \leq i \leq |C|$，令 $\alpha'(sat_i) := 0$ 。此外，对于每个 $1 \leq j \leq 3$ 和任意 $x \in S$，以及 $b \in [\![x]\!]_j$，(iii) 我们令 α'(var(b)) := sgn(b)，(iv) 同时令 $\alpha'(val_j) := \alpha(x)$。然后，根据构造（i）

对于 α'，根据公式（9），我们有 $D'[\alpha'] \neq \emptyset$。由于 $D[\alpha] \neq \{\emptyset\}$，尽管采用了构造（i），$\alpha'$ 仍不满足公式（7）中的任何一个。此外，根据构造（ii），α' 也不满足公式（3）–（6）中的任何一个。最后，根据构造（iii）和（iv），α' 不满足公式（1）或（2）中的任何一个。因此，$C'[\alpha'] \neq \{\emptyset\}$，由此完成此情形的证明。 □

定理 IV.5（稀疏反馈顶点集的下界）。给定一个量词深度为 ℓ 的任意合取析取范式量化布尔公式 Q，以及 Q 的一个最小稀疏反馈顶点集 S，其满足 $k = |S|$。那么，根据指数时间假设，QSAT$_\ell$ 在 Q 上无法在 tow$(\ell, o(k)) \cdot$ poly(|var(Q)|) 时间内判定。

证明。无需 loss of generality，我们假设 Q 属于 3,1- 合取析取范式。量词深度为 $\ell = 1$ 的结果对应于 SAT，因此由指数时间假设立即可得，因为 $k \leq$ |var(Q)|，且指数时间假设意味着不存在一种算法，能够在 $2^{o(|\text{var}(Q)|)}$ 时间内求解 3- 合取范式。对于 $\ell > 1$ 的情形，我们采用归纳法。假设该结果对 $\ell-1$ 成立。令 Q 为一个量词深度为 $\ell - 1$ 的量化布尔公式。

情形 1：Q 的最内层量词 $Q_{\ell-1}$ 为 $\exists$。此时，我们对 (Q, S) 应用归约 $\mathcal{R}$。由此，我们在 $\mathcal{O}$(poly(|var(Q)|) 时间内得到一个所得实例 Q' 和 $G_{Q'}$ 的一个反馈顶点集 S'，参见定理 IV.2。根据定理 IV.3，该归约是正确的；也就是说，matr(Q) 的满足赋值集合与 matr(Q') 的满足赋值集合在限制到变量 var(Q) 后相同。此外，根据引理 IV.1，有 $|S'| \leq 3 \lceil \log(|S|) \rceil$。反设尽管指数时间假设成立，仍可在 tow$(\ell, o(\log(k))) \cdot$ poly(|var(Q')|) 时间内判定 Q' 上的 QSAT。那么，就可以在 tow$(\ell - 1, o(k)) \cdot$poly(|var(Q)|) 时间内判定 Q 的有效性，这与该假设矛盾。

情况 2：Q 的最内层量词 $Q_{\ell-1}$ 是 $\forall$。在这种情况下，我们采用与情形 1 类似的方法，但首先反转 Q，从而得到量化布尔公式 $Q^\star$，其量词块被翻转，使得 matr($Q^\star$) 再次处于 3,1-CDNF 中。然后，与情形 1 一样，在对 ($Q^\star$, S) 应用归约 $\mathcal{R}$ 后，我们得到 Q'。下界的推导与上文类似，因为只需用常数时间即可反转 Q' 的真值。 □

推论 IV.6（关联反馈顶点集的下界）。给定任意量化布尔公式 Q，其 $F =$matr(Q) 处于合取范式（析取范式）中，且 Q 的最内层量词 Q_ℓ 为$Q_\ell = \exists$ （$Q_\ell = \forall$），其中 I_F 的反馈顶点数为 k。那么，在指数时间假设下，不能在时间 tow$(\ell, o(k)) \cdot$ poly(|var(Q)|) 内判定 Q 上的 QSAT$_\ell$。

证明。我们证明，任意 3,1-CDNF 中的量化布尔公式 Q' 的稀疏反馈顶点数，均以线性方式界定 I_F 的反馈顶点数，其中 F 是根据 Q' 的最内层量词，将 matr(Q') 转换为合取范式或析取范式后得到的。令 $Q' = Q_1 V_1 . \cdots Q_\ell V_\ell . F'$ 为任意一个 3,1-CDNF 中的量化布尔公式，令 S' 为一个最小稀疏反馈顶点集，其中 Q' 的 $k' = |S'|$。我们构造量化布尔公式 $Q := Q_1 V_1 . \cdots Q_\ell V_\ell . F$，其中 F 的定义如下。

情形 1：$Q_\ell = \exists$ 因此 $F' = C \wedge D$。我们定义 $F := C \cup \{f\}$ 为合取范式，其中 $f := \{l \mid \{l\} \in D\}$ 是一个子句。容易看出，$S := S' \cup \{f\}$是 I_F的反馈顶点集。反设 $I_F - S$中存在一个环。根据关联图 I_F的定义，该环在 var(F) $\setminus S$ 和 $F \setminus S$中的顶点之间交替出现，即我们有以下两种情况之一：(1) 限制在 var(F) 中的环已经存在于 G_Q中，这与 S是 G_Q的一个（稀疏）反馈顶点集相矛盾；或 (2) 该环的形式为 x、c、y、c'、x，其中 c和 c'是不同的子句，这与 S 是稀疏的这一假设相矛盾。因此，假设指数时间假设成立，并且我们可以在 tow$(\ell, o(|S|)) \cdot$ poly(|var(Q)|) 时间内判定 Q 的有效性，这将与定理 IV.5 相矛盾。

情形 2：$Q_\ell = \forall$，即 $F' = D \vee C$。我们定义 $F := D \cup \{f\}$，其析取范式为 $f := \{l \mid \{l\} \in C\}$，并按照情形 1 继续。 □

注意，可以在时间 $2^{\mathcal{O}(k)} \cdot$poly(|var(G)|) 内求得图 G 的最小反馈顶点集，其中 k 是图 G [41, 的反馈顶点数。对于计算 Q 的最小稀疏反馈顶点集，可以采用更强的参数树宽 k'，改造标准反馈顶点集动态规划算法；该参数可在时间 $2^{\mathcal{O}(k')} \cdot$poly(|var($G_Q$)|) [37] 内得到 2-近似。这些算法可以改造用于稀疏反馈顶点集，并在时间 $2^{\mathcal{O}(k' \cdot \log(k'))} \cdot$poly(|var($G_Q$)|) [13] 内运行。

众所周知，将 QSAT 限制在 2-CNF 公式 [2]时，可在多项式时间内求解。作为我们归约 $\mathcal{R}$的一个应用，我们现在指出，即使只允许出现一个任意长度的子句，也会导致问题变得不可解。

推论 IV.4. 量化布尔公式上的 QSAT 问题 $Q = Q_1 V_1 \cdots Q_\ell V_\ell . C \wedge D$，其量词深度为 $\ell \geq 2$，其中 $Q_\ell = \exists$、C 属于合取范式 2-CNF，且 D 属于 1-DNF，是 $\Sigma^{\text{P}}_{\ell-1}$-完全的（若 $Q_1 = \exists, \ell$ 为奇数），且是 $\Pi^{\text{P}}_{\ell-1}$-完全的（若 $Q_1 = \forall, \ell$ 为偶数）。

证明（概要）。我们仅概述 Q_1=$\exists$ 的证明（ Q_1=$\forall$ 的情形证明类似）。对于成员性，只需注意到 $C \wedge D$ 的可满足性可以在多项式时间内判定。我们只需针对每个文字 $l \in D$，尝试将 l 赋值为真，判断由此得到的 2-CNF 公式是否可满足。对于困难性，设 $Q' = \forall V'_1 \cdots \exists V'_{\ell-1} . C'$ 是一个具有 $\Pi_{\ell-1}$ 前缀且 C' 为（3-）合取范式的量化布尔公式。应用该归约，我们得到一个等价的量化布尔公式 $\overline{Q} = \mathcal{R}(Q', \emptyset)$，实际上只使用了公式（3）、（8）、（9）。量化布尔公式 $\overline{Q}$ 的矩阵为 $C \vee D$，其中 C 属于 2-DNF，D 属于 1-CNF。通过对 $\overline{Q}$ 取反（并翻转量词类型），我们得到一个矩阵为 $C \vee D$ 的量化布尔公式 Q，其中 C 属于 2-CNF，D 属于 1-DNF。

由于 Q' 的选取是任意的，判定 $\overline{Q}$ 是 $\Pi^P_{\ell-1}$-困难的，因此判定 $Q = \neg\overline{Q}$ 是 $\Sigma^{\mathrm{P}}_{\ell-1}$-困难的。 □

B. 树深度的困难性见解与新下界

该归约 $\mathcal{R}_{\mathsf{td}}$ 适用于受限的树深度分解，我们称之为 α- 树深度分解。对于给定的整数 α，一个 α- 树深度分解 T由一条路径 S组成，该路径称为主路径，其中每个节点$v \in S$ 都附着一棵高度至多为 α 的树。这些树的每个叶节点 r 又连接到一条高度至多为 $|S|$的路径 P_r。令 $\mathcal{P}$ 表示这些路径的集合。

除了 $\mathcal{R}$引入的变量外，我们还需要以下变量集合，因此假设存在一个针对 Q的量词深度 ℓ 的 ℓ-树深度分解 T of G_Q。对于每个 $P_r \in \mathcal{P}$，我们都需要类似于 VarIdxs 的新索引变量，这些变量在 $\mathcal{P}$中的不同路径之间共享。我们令 VarPIdxs : $= \{idx_j^{\mathbb{P},1}, \ldots, idx_j^{\mathbb{P},\lceil \log(\max_{P_r\in\mathcal{P}}|P_r|\rceil} \mid 1 \le j \le 3\}$。这些 $2^{\lceil \log(\max_{P_r\in\mathcal{P}}|P_r|\rceil}$ 个每个索引对应的变量组合足以寻址 P_r中的任意 $|P_r|$ 个元素。与上文类似，我们为每个元素 $x \in P_r$ 以及每个 $1 \le j \le 3$ 分配一个集合，该集合由这些索引变量 $idx^{\mathbb{P}}_{,j}{}^{1}, \ldots, idx^{\mathbb{P}}_{,}{}^{\lceil \log(\max_{P_r\in\mathcal{P}}|P_r|\rceil}_{j}$ 上的任意但固定的文字组合构成（在 ${}^{j}P_r$内唯一），我们将其记为 $[\![x]\!]^{\mathbb{P}}_j$。此外，我们还需要三个布尔变量 VarPVals : $= \{val_1^{\mathbb{P}}, val_2^{\mathbb{P}}, val_3^{\mathbb{P}}\}$，其中 $val_j^{\mathbb{P}}$捕获通过 VarPIdxs 中的变量寻址的元素在第 j个索引处的真值。最后，为了为这些索引变量设置上下文，我们定义一个选择变量集合 VarPSels $\coloneqq \{sel_r^{\mathbb{P}} \mid P_r \in \mathcal{P}\}$，其中 $sel_r^{\mathbb{P}}$ 表示 VarPIdxs 中的变量是否用于寻址 P_r中的元素。

该归约 $\mathcal{R}_{\mathsf{td}}$ 接收 Q 和 T（由此产生 S 和 $\mathcal{P}$），并构造一个实例 Q' 以及一个关于 Q'的树深度分解 T'。所构造的量化布尔公式等于 $Q' \coloneqq \exists V_1.\ \forall V_2.\ \cdots \exists V_\ell.\ \forall$VarIdxs、VarVals、VarSat、VarPIdxs、VarPVals、VarPSels . $C' \vee D'$，其中 C' 为析取范式，定义为以下项的析取:

$$x \wedge \bigwedge_{b\in[\![x]\!]_j} b \wedge \neg val_j \qquad \text{for each } x \in S, 1 \le j \le 3 \tag{1}$$

$$\neg x \wedge \bigwedge_{b\in[\![x]\!]_j} b \wedge val_j \qquad \text{for each } x \in S, 1 \le j \le 3 \tag{2}$$

$$sat_i \wedge l \qquad \text{对于每个满足以下条件的 } c_i \in C、\ 1 \le j \le 3 \ \mathsf{lit}(c_i, j) = l, \mathsf{var}(l) \in \mathsf{var}(C) \setminus (\bigcup_{P_r\in\mathcal{P}} P_r \cup S) \tag{12}$$

$$sat_i \wedge \neg b \qquad \text{对于每个 } c_i \in C, 1 \le j \le 3, x \in S, b \in [\![x]\!]_j\text{，满足 } \mathsf{var}(\mathsf{lit}(c_i, j)) \text{ 的 } = x \tag{4}$$

$$sat_i \wedge val_j \qquad \text{对于每个 } c_i \in C、\ 1 \le j \le 3、\ x \in S \text{ 满足 } \mathsf{lit}(c_i, j) = x \tag{5}$$

$$sat_i \wedge \neg val_j \qquad \text{对于每个 } c_i \in C、\ 1 \le j \le 3、\ x \in S \text{ 其中 } \mathsf{lit}(c_i, j) = \neg x \tag{6}$$

$$sat \wedge l \qquad \text{for each } \{l\} \in D \tag{7}$$

$sel_r^{\mathbb{P}} \wedge x \wedge \bigwedge_{b \in [\![x]\!]_j^{\mathbb{P}}} b \wedge \neg val_j^{\mathbb{P}}$对于每个 $P_r \in \mathcal{P}, x \in P_r, 1 \leq j \leq 3$

(13)$sel_r^{\mathbb{P}} \wedge \neg x \wedge \bigwedge_{b \in [\![x]\!]_j^{\mathbb{P}}} b \wedge val_j^{\mathbb{P}}$对于每个 $P_r \in \mathcal{P}, x \in P_r$, $1 \leq j \leq 3$(14)$sat_i \wedge \neg b$对于每个 $c_i \in C, 1 \leq j \leq 3, P_r \in \mathcal{P}$, $x \in P_r, b \in [\![x]\!]_j^{\mathbb{P}}$，满足 var(lit($c_i, j$)) 的 $= x$ (15)

$sat_i \wedge val_j^{\mathbb{P}}$对于每个 $c_i \in C, 1 \leq j \leq 3, P_r \in \mathcal{P}, x \in P_r$，满足 lit($c_i, j$) 的 $= x sat_i \wedge \neg val_j^{\mathbb{P}}$对于每个 $c_i \in C, 1 \leq j \leq 3$, $P_r \in \mathcal{P}, x \in P_r$，满足 lit($c_i, j$)$= \neg x sat_i \wedge \neg sel_r^{\mathbb{P}}$对于每个 $c_i \in C, P_r \in \mathcal{P}$，var($c_i$) $\cap P_r \neq \emptyset$ (16)

(17)

(18)

与上文类似，我们在 1-CNF 中定义 D'，它是以下单元素集合的合取。

$$\neg sat_i \qquad \text{对每个 } 1 \leq i \leq |C| \tag{8}$$

$$\neg sat \tag{9}$$

由此可见，我们基本上重用了公式 (1)– (7)，因为公式 (12) 只是对 (3) 的轻微修改。公式 (13) 和 (14) 的作用方式与 (1) 和 (2) 类似，但它们会在 $P_r \in \mathcal{P}$ 中同步路径上变量的索引和值，因此还取决于用于表示 P_r 为真的选择器 $sel_r^{\mathbb{P}}$。接着，公式 (15)– (17) 的作用方式与 (4)–(6) 类似，但针对特定路径 $P_r \in \mathcal{P}$ （而非 S）。此外，我们只考虑这样的情形：在判定子句 $c_i \in C$ 是否可满足（sat_i）时，如果 P_r 包含相关变量，则对应于路径 $P_r \in \mathcal{P}$ 的选择器 $sel_r^{\mathbb{P}}$ 也成立。公式 (18) 确保了这一点。

请注意，定理 IV.2 中所述的运行时间对于 $\mathcal{R}_{\mathsf{td}}$ 同样成立。此外，正确性的证明也与定理 IV.3 中的证明非常相似，正如下文所述。

命题 XI.2（正确性）。给定一个量词深度为 ℓ 的 QBF Q，其矩阵 matr(Q) $= C \wedge D$ 属于 3,1-CDNF，以及一个 ℓ-树深度分解 T，该分解由 G_Q 构成，包含主路径 S 和路径 $\mathcal{P}$。随后，归约 $\mathcal{R}_{\mathsf{td}}(Q, T)$ 计算出一个实例 Q'，该实例与 Q 等价。事实上，对于将变量 α 赋值给 matr(Q) 中变量的任意赋值，如果且仅如果将 α 扩展为变量 VarIdxs ∪VarVals ∪ VarSat ∪VarPIdxs ∪VarPVals ∪VarPSels Q' 的每个扩展 α' 都满足 matr(Q)，则该赋值满足 matr(Q)。

证明（概要）。注意，正确性证明与定理 IV.3 的证明几乎完全相同。VarPIdxs 中的索引变量及其对应的值 VarPVals，行为方式类似于集合 S 的 VarIdxs 和 VarVals。唯一的区别在于额外的选择器变量 VarPSels。关键在于，公式 (18) 确保当 sat_i 被设为 1 时，子句 c_i 的可满足性自然成立，但路径 $P_r \in$ 对应的选择变量 $sel_r^{\mathbb{P}}$ $\mathcal{P}$ 中包含的变量 c_i 被设为 0。请注意，将多个选择器变量设为 1 也不会造成问题，因为与只设一个变量相比，这样可能会使更多公式 (13) 和 (14) 得到满足。 □

下界所缺少的要素是结构感知性，具体形式化如下。

引理 XI.3（递减树深度）。给定一个量词深度为 ℓ 的 QBF Q，其矩阵 matr(Q) $= C \wedge D$ 属于 3,1-CDNF，以及一个 ℓ-树深度分解 T，该分解由 G_Q 构成，包含主路径 S 和路径 $\mathcal{P}$，并满足 $h = |S|$。随后，归约 $\mathcal{R}_{\mathsf{td}}$ 基于 Q 和 T 构造一个 QBF $Q' = \mathcal{R}_{\mathsf{td}}(Q, T)$，并为 $G_{Q'}$ 构造一个高度为 $\mathcal{O}(\log(h)+\ell)$ 的 $(\ell+1)$-树深度分解 T'。

证明。我们展示 $\mathcal{R}_{\mathsf{td}}$ 如何产生一个高度为 $\mathcal{O}(\log(h)+\ell)$ 的（$\ell+1$）树深度分解 T'，参见图 8，其中描绘了 $\mathcal{P} = \emptyset$ 的特殊情形。为此，令 $S' :=$ VarIdxs ∪VarVals ∪$\{sat\}$∪ VarPIdxs ∪VarPVals 为通过 $\mathcal{R}_{\mathsf{td}}$获得的辅助变量。我们构造 $T' :=$ (var(Q') F')，其中假设顶点 s_i 在 S' 中具有任意固定的全序 $s_1, ..., s_{|S'|}$，并满足 $1 \leq i \leq |S'|$。因此，令 F' 如下：(i) 对于每个 $1 \leq i \leq |S'|$，定义 (s_i, s_{i+1}) $\in F'$，即 S' 的顶点彼此相连。此外，(ii) 将 S 中的变量与 S' 中的变量相连，其中对于每个 $s \in S$，令 ($s_{|S'|}, s$) $\in F'$。然后，(iii) 保留 var(Q) $\setminus$ ($\bigcup_{P_r \in \mathcal{P}} P_r \cup S$) 内的边，即对于每个 $u, v \in$var(Q)$\setminus$($\bigcup_{P_r \in \mathcal{P}} P_r \cup S$)，若 ($u$, v) $\in F$，则令 (u, v) $\in F'$。进一步地，(iv) 对于每个 $v \in$ var(Q) $\setminus$ ($\bigcup_{P_r \in \mathcal{P}} P_r \cup S$)，若对于某个 $s \in S$ 有 (s, v) $\in F$，则构造 ($s_{|S'|}, v$) $\in F'$。我们将 VarPSels 中的变量 (v) 与 T' 相连，其中对于每个 $P_r \in \mathcal{P}$ 和 $x \in P_r$，分别定义 (r, $sel_r^{\mathbb{P}}$) $\in F'$ 以及 ($sel_r^{\mathbb{P}}, x$) $\in F'$。最后，(vi) VarSat 中的变量将在 F' 中成为新叶节点。对于每个满足 $c_j \in C$ 的 $sat_j \in$ VarSat，我们在 F' 中定义 $s_{|S'|}$ 的唯一后代节点：(1) 如果对于某个 $P_r \in \mathcal{P}$ 有 var(c_j) $\cap P_r \neq \emptyset$，则要求 ($sel_r^{\mathbb{P}}, sat_j$) $\in F'$；(2) 如果 var(c_j) $\subseteq S$，则定义 ($s_{|S'|}, sat_j$) $\in F'$；(3) 否则，令 $x \in$ var(c_j) $\setminus S$ 为满足 var(c_j) $\setminus S$ 的变量，其中仅包含 x 以及 x 在 F 中的祖先；然后定义 (x, sat_j) $\in F'$。注意，由于 T' 的构造，T' 的高度被 $\mathcal{O}(\log(h)+\ell)$ 所界定，因为 $|S'| \in \mathcal{O}(\log(h))$，并且由于 $T' - S'$ 的高度属于 $\mathcal{O}(\log(h) + \ell)$，这是因为对于每个 $P_r \in \mathcal{P}$ 都有 $|P_r| \in \mathcal{O}(h)$。此外，注意，将公式 (1) 和 (2) 中的每个项按照第 IV-A 节所述从 DNF 规范化为 3-析取范式，会为每个 $x \in S$ 增加两条高度为 $\mathcal{O}$ (log(h)) 的路径。类似地，将公式 (13) 和 (14) 中的项从 DNF 转换为 3-析取范式，会为每个满足 $P_r \in \mathcal{P}$ 的 $x \in P_r$ 增加两条高度为 $\mathcal{O}$(log(h)) 的路径。由于 $\bigcup_{P_r \in \mathcal{P}} P_r \cup S$ 中的变量不处于祖先/后代关系

中，规范化保留了 $T'(\mathcal{O}(\log(h)+\ell))$ 的高度。最后，注意 T' 确实是 $G_{Q'}$ 的一个定义良好的（ℓ+1）树深度分解，并且可以如下验证 $\mathcal{R}_{\mathsf{td}}$ 的公式。公式 (1) 和 (2) 由 (i) 和 (ii) 处理，即这些项中的变量通过 (i) 和 (ii) 得以保证在 T' 中处于祖先/后代关系。通过 (iii)、(iv) 和 (vi)，我们确保公式 (12) 中的变量在 T' 中互为祖先 / 后代。注意，公式 (8)–(9) 不会在 $G_{Q'}$ 中产生边。然后，公式 (4)–(6) 由 (i) 和 (vi) 处理，公式（7）由 (i)–(v) 处理。公式 (13) 和 (14) 由 (i) 和 (v) 覆盖。此外，公式 (15)–(17) 也由 (i) 和 (vi) 处理。最后，公式 (18) 由于 (v) 被 T' 覆盖，从而完成证明。 □

定理 IV.7（ ℓ-树深度分解的下界）。给定任意一个量词深度为 ℓ、属于合取析取范式的量化布尔公式 Q，以及一个高度为 $k=$、对应 G_Q 的 ℓ-树深度分解 T，其中高度记为 $\mathsf{td}(G_Q)$。那么，在指数时间假设成立的条件下，无法在时间 $\mathsf{tow}(\ell, o(k-\ell))\cdot\mathrm{poly}(|\mathrm{var}(Q)|)$ 内判定 Q 上的 QSAT_ℓ 。

证明。无妨设 Q 属于 3,1- 合取析取范式。量词深度为 $\ell=1$ 的情形对应于 SAT，因此由指数时间假设立即可得结论，因为 $k\le|\mathrm{var}(Q)|$ ，且指数时间假设意味着不存在能在时间 $2^{o(|\mathrm{var}(Q)|)}$ 内求解 3-CNF 上 SAT 的算法。对于 $\ell>1$ 的情形，我们使用归纳法。假设结论对 $\ell-1$ 成立。

情形 1： Q 的最内层量词 Q_ℓ 是 $\exists$。此时，我们对 (Q,T) 应用归约 $\mathcal{R}_{\mathsf{td}}$，其中 S 是 T 的主路径。由此，我们在时间 $\mathcal{O}(\mathrm{poly}(|\mathrm{var}(Q)|)$ 内得到一个所得实例 Q'，以及一个 $G_{Q'}$ 的树深度分解 T'，参见定理 IV.2。根据命题 XI.2，该归约是正确的：当限制在变量 $\mathrm{var}(Q)$ 上时，$\mathrm{matr}(Q)$ 的满足赋值集合与 $\mathrm{matr}(Q')$ 的满足赋值集合相同。此外，由引理 XI.3 可知，T' 的高度 k' 有界于 $\mathcal{O}(\log(|S|)+\ell)$。反设尽管指数时间假设成立，仍能在时间 $\mathsf{tow}(\ell, o(k'-\ell))\cdot\mathrm{poly}(|\mathrm{var}(Q')|)$ 内判定 Q' 上的 QSAT。于是，可以在时间 $\mathsf{tow}(\ell-1, o(k))\cdot\mathrm{poly}(|\mathrm{var}(Q)|)$ 内判定 Q 的有效性，这与归纳假设矛盾。

情况 2： Q 的最内层量词 Q_ℓ 是 $\forall$。我们采用与情形 1 类似的方法，但首先反转 Q，得到一个量化布尔公式 $Q^\star$，其量词块顺序经过翻转，使得 $\mathrm{matr}(Q^\star)$ 属于 3,1-CDNF。随后，与情形 1 一样，对 $(Q^\star, S)$ 应用归约 $\mathcal{R}_{\mathsf{td}}$ 后，我们得到 Q'。下界的推导与上文类似，因为只需在常数时间内反转 Q' 的有效性即可。 □

观察 XI.4. 存在一个常数 c，使得对于任意整数 $\ell\ge1$，都有 $\mathsf{tow}(\lfloor\log(\ell)\rfloor, 0)\ge c\cdot\ell$。

推论IV.8（树深度的下界）。存在一个线性函数 f 和一个整数 $\ell_0>0$，使得对于任意给定的量词深度为 $\ell\ge\ell_0$、属于 3,1-CDNF 的量化布尔公式 Q，以下结论成立：在指数时间假设成立的条件下，无法在时间 $\mathsf{tow}(f(\ell)\ o(\mathsf{td}(G_Q)))\cdot\mathrm{poly}(|\mathrm{var}(Q)|)$ 内判定 QSAT_ℓ on Q 。

证明。令 $k:=\mathsf{td}(G_Q)$。定理 IV.7 推出 (A)：在指数时间假设成立的条件下，无法在时间 $\mathsf{tow}(\ell, o(k-\ell))\cdot\mathrm{poly}(|\mathrm{var}(Q)|)$ 内求解 QSAT_ℓ on Q 。我们分情况讨论。情况 $\ell\le g(k)$，其中 g 是线性函数：应用推论 (A)，可得 $f(\ell):=\ell$，因为 $o(k-\ell)=o(k)$。情况 $\ell>g(k)$：与前一种情况类似，我们应用推论 (A)。由观察 XI.4 可得，对于某个常数 c，有 $\mathsf{tow}(\lfloor\log(\ell)\rfloor, 0)\ge c\cdot\ell$，因此 (i) $\mathsf{tow}(\lfloor\log(\ell)\rfloor, 0)\in\Omega(k)$，因为根据情况假设，$k<g^{-1}(\ell)$，且 g^{-1} 是线性函数。此外，即使 $\ell\gg k$，我们也有 (ii) $2^{k-\ell}>0$。结合 (i) 和 (ii)，可得 $f\in\Omega(\ell-\lfloor\log(\ell)\rfloor-1)$，从而得到所需结论：对于充分大的 ℓ_0，f 是线性的。进而，这就证明了该结论。 □

推论 IV.9（关联树深度的下界）。给定任意一个 Q 含有 $F=\mathrm{matr}(Q)$ 的合取范式（析取范式）量化布尔公式，使得其最内层量词 Q_ℓ 为 Q $Q_\ell=\exists$（$Q_\ell=\forall$），且 $k=\mathsf{td}(I_Q)$。则在指数时间假设下，QSAT_ℓ on Q 无法在时间 $\mathsf{tow}(\ell, o(k-\ell))\cdot\mathrm{poly}(|\mathrm{var}(Q)|)$ 内判定。

证明。$\ell=1$ 的情形可由指数时间假设立即推出，因此我们假设 $\ell>1$。我们证明，任意处于 3,1-CDNF $Q'=Q_1V_1.\cdots Q_\ell V_\ell.F'$ 中的量化布尔公式的树深度，都线性地界定 I_F的树深度，其中 F是通过将 $\mathrm{matr}(Q')$ 转换为合取范式或析取范式而得到的（定义与推论 IV.6 中完全相同）。首先，我们在时间 $2^{\mathcal{O}(k^2)}\cdot\mathrm{poly}(|\mathrm{var}(F')|$ [49] 内计算 $G_{Q'}$的一个树深度为 k的树深度分解 $T'=(\mathrm{var}(F')\ E')$。然后，我们构造一个量化布尔公式 $Q:=Q_1V_1.\cdots Q_\ell V_\ell.F$，其中 F定义如下（见推论 IV.6）。

情形 1： $Q_\ell=\exists$ 因此 $F'=C\wedge D$。我们定义 $F:=C\cup\{f\}$ 为合取范式，其中 $f:=\{l\mid\{l\}\in D\}$ 是一个长子句。接下来，我们对 T'进行略微调整，得到 I_F的树深度分解 $T=(\mathrm{var}(F')\cup C\cup\{f\}, E'\cup E)$，其中 E定义如下。首先，f是 T的新根节点，即对于 T' 的根 r，我们定义$(f,r)\in E$。此外，对于每个子句 $c\in C$和变量 $v\in\mathrm{var}(c)$，如果$\mathrm{var}(c)$中的每个其他变量都是 v 在 T'中的祖先，我们添加一条边$(v,c)\in E$。注意到 T的高度有 $k+2$的上界。因此，假设指数时间假设成立，并且我们可以在时间$t\ell, o(k-\ell)\cdot\mathrm{poly}(|\mathrm{var}(Q)|)$内判定 Q 的有效性，这将与定理 IV.7矛盾。

情况 2： $Q_\ell = \forall$，即 $F' = D \vee C$。我们定义 $F := D \cup \{f\}$ 为析取范式，其中 $f := \{l \mid \{l\} \in C\}$，并按照情形 1 中的方法进行。 □

第十二节. 第五节（使用顶点覆盖数和反馈边数求解 CQSAT 的算法）的省略证明

定理 V.1. 给定任意一个 QSAT 的合取范式量化布尔公式 Q，其中 k 是 I_Q 的顶点覆盖数。那么，可以在 $1.709^{3^k} \cdot \text{poly}(|\text{var}(Q)|)$ 时间内判定 Q 的有效性。

证明. 设 C 是 I_Q 的一个顶点覆盖，其中包含 k_1 个变量和 k_2 个子句。那么， Q 至多包含 $3^{k_1} + k_2$ 个不同的子句。这是因为， C 之外每个子句中的变量都必须是 C 的子集，而其他所有子句都包含在 C 中。由此可得该引理，因为每个含有 m 个子句的合取范式量化布尔公式都可以在 $\mathcal{O}(1.709^m)$ [65] 时间内求解。 □

现在，我们从定理 V.2 开始，对定理 V.2 和定理 V.5 给出正式证明。我们的核化首先使用命题 VIII.2 计算 G_Q 的最小反馈边集 D。我们用 $V(D)$ 表示与 D 中的边关联的顶点集合。令 H 为图 $G_Q - D$。那么， H 是 G_Q 的生成森林。

我们先介绍一些适用于任意合取范式量化布尔公式的简单归约规则。特别地，我们注意到，可以假设 Q 不含单位子句（即恰好包含一个文字的子句），也不含纯文字（即在 Q中只以正形式出现或只以负形式出现的变量）。为证明前者，假设 Q 包含一个关于某个变量 v的单位子句 C。如果 v 是全称玩家，那么 Q 不可满足，因此我们可以返回一个常数规模的不可满足公式作为其核。另一方面，如果 v 是存在玩家，那么 Q等价于公式 $Q[v = b]$；其中，若 v在 C 中以正形式出现，则 b 为1；若 v 在C中以负形式出现，则 $b = 0$ 为1。请注意，在这两种情况下，反馈边集的大小都不会增加（因为 $G_{Q[v=b]}$ 是 G_Q的子图），且公式中的变量数会减少。

请注意， H 中不属于 V（D）的每个顶点 v 都只出现在大小至多为2的子句中。此外，由于 Q 不再包含单位子句，因此每个这样的顶点 v 只出现在大小恰为2的子句中。

现在我们给出一系列归约规则，以便得到一个约化实例，其中 u 和 v 恰好共同出现在一个大小为二的子句中。下面的引理使我们能够处理 u 和 v 共同出现在 3 个或 4 个子句中的情形。

引理 XII.1. 设 $e = \{u, v\}$ 是 H 的一条干净边。那么：

(1) 如果 u 和 v 包含在 Q的 4 个子句中，则 Q不可满足。

(2) 如果 u 和 v 包含在 Q的 3 个子句中，则 Q 不可满足，或者存在一个赋值 τ： $\{u, v\} \to \{0, 1\}$，使得 Q 等价于 $Q[\tau]$。

证明。性质 (1) 是显然的，因为如果 u 和 v 包含在 Q 的四个子句中，那么不存在 u 和 v 的赋值能够满足这四个子句，因此 Q 不可满足。

为证明性质 (2)，假设 u 和 v 包含在 Q 的恰好三个子句中。那么，存在唯一赋值 τ: $\{u, v\} \to \{0, 1\}$ 满足这三个子句。因此，如果变量 u 和 v 中有一个是全称玩家，则 Q 不可满足。此外，在其他情况下，两个变量都是存在玩家，因此 Q 与 $Q[\tau]$ 等价。

□

注意到，引理 XII.1，特别是对 H 的每条干净边穷尽地应用该引理，表明我们可以在多项式时间内计算出一个常数大小的核，或者计算出一个等价实例 Q'，使得 $G_{Q'}$ 是 G_Q 的子图，并且 H 的每条干净边的端点共同包含在至多 2 个子句中。因此，在下文中，我们可以假设 H 的每条干净边的端点共同出现在 Q 的至多 2 个子句中。下面的引理进一步强化了这一点，使我们可以假设 H 的每条干净边的端点共同恰好出现在 Q 的一个子句中。

在介绍该引理之前，我们需要以下记号，以便用另一个文字替换（代换）某个文字。设 l 和 l' 是定义在不同变量上的两个文字。我们将 $Q[l \to l']$ 定义为如下公式：从 Q 出发，将 Q 的矩阵中 $l(\neg l)$ 的每次出现替换为 l' $(\neg l')$，并从 Q 的前缀中移除 l 的变量。

引理 XII.2. 设 $e = \{u, v\}$ 是 H的一条干净边，且 u 和 v 在 Q的恰好 2 个子句中共同出现。那么，以下情况之一成立：

- Q 不可满足，或者
- 存在一个关于 u 或 v 的赋值 τ，使得Q 等价于 $Q[\tau]$，或者
- Q 等价于 $Q[l \to l']$，其中 l 和 l' 是关于 u 和 v 的文字，而 l 和 l' 则分别属于不同的变量。

证明。由于 u和 v恰好共同包含于 Q的 2 个子句中（且这些子句不包含任何其他变量），不妨设这些子句为 C_1 和 C_2，因此恰好有两个赋值 τ_1： $\{u, v\} \to \{0, 1\}$和 τ_2： $\{u, v\} \to \{0, 1\}$，能够满足这两个子句。我们区分以下情况。如果存在一个变量 $w \in \{u, v\}$ 使得 $\tau_1(w) = \tau_2(w)$，那么要满足 C_1 和C_2，唯一的方法就是将变量 w 设为 $\tau_1(w)$。因此，如果 w是全称玩家，则 Q 不可满足。此外，如果 w 是

存在玩家，那么 Q 等价于 $Q[\tau]$，其中 τ 是将 w 设为 $\tau_1(w)$ 的赋值。

否则，有 $\tau_1(u) \neq \tau_2(u)$ 以及 $\tau_1(v) \neq \tau_2(v)$。因此，对于每个 i with $1 \leq i \leq 2$，要么 $\tau_i(u) = \tau_i(v)$，要么 $\tau_i(u) = \neg\tau_i(v)$。由于这两种情况相似，我们只证明前一种情况的陈述（即 $\tau_i(u) = \tau_i(v)$）。不失一般性，设 v 是在 Q 的前缀中出现在 u 后面的变量。如果 v 是全称玩家，则 Q 不可满足，因为全称获胜策略可以确保 v 被赋予与 u 不同的值。另一方面，如果 v 是存在玩家，则我们断言 Q 等价于 $Q[v \to u]$。为证明正向方向，设 $\mathcal{T}$ 是 Q 的存在玩家获胜策略，并令 δ 为全称玩家的任意一局对局。于是，$\alpha(\mathcal{T}, \delta)$ 满足包含 u 和 v 的两个子句，因此 $\alpha(\mathcal{T}, \delta)[u] = \alpha(\mathcal{T}, \delta)[v]$，这表明 $\alpha(\mathcal{T}, \delta)$ 也满足 $Q[v \to u]$。另一方面，如果 $\mathcal{T}$ 是 $Q[v \to u]$ 的存在玩家获胜策略，那么，将 v 设为 u 的值而从 $\mathcal{T}$ 获得的策略 $\mathcal{T}'$ 显然是 Q 的存在玩家获胜策略；这是可行的，因为在 Q 的前缀中，v 出现在 u 之后。 □

因此，根据引理 XII.2 和引理 XII.1，我们从现在起可以假设，H 的每条干净边的两个端点恰好同时出现在 Q 的一个子句中（且该子句不包含其他变量）。接下来的引理允许我们移除包含在 H的干净边中的叶节点。

引理 XII.3. 设 l 是 H 和 $l \notin V(D)$ 的一个叶节点。那么，存在一个赋值 $\tau\colon \{l\} \to \{0,1\}$，使得 Q 与 $Q[\tau]$等价。

证明。由于 l 是 H、$l \notin V(D)$ 的一个叶节点，并且我们可以假设 Q 不包含任何单位子句，因此 l 只出现在与其父节点 p 一起出现于 H中的子句里。还要注意，$e = \{p, l\}$ 是 H 的一条干净边，因为 l 是一个叶节点，且 $l \notin V(D)$。因此，l 恰好包含在 Q的一个子句 C 中（与 p一起）。但是，如果 l 是存在玩家（全称玩家），那么 Q 就等价于公式 $Q[l = b]$（$Q[l = \neg b]$），其中，如果 l 在 C 中以正号出现，则 b 为 1；如果 l 在 Q中以负号出现，则 $b = 0$ 为 1。 □

注意，在穷尽地应用引理 XII.3 后，我们得到一个实例，使得 H 的每个叶节点都是 D 中边的至多 $2|D|$ 个端点之一。因此，H 至多有 $2|D|$ 个叶节点，从而至多有 $2|D|-2$ 个度至少为 3 的顶点。因此，现在只需减少 H 中度恰好为 2 的顶点数量，我们将通过以下观察和引理来实现这一点。

我们称 P的 H 中的一条路径为干净路径，当且仅当 P 的所有内部顶点在 H 中的度为二，且不属于 $V(D)$。

观察 XII.4。设 P 是 H 中一条至少具有两个内部顶点的干净路径。那么，P 的每条边都是干净的。此外，如果 v 是 P 的一个内部顶点，则 v 在 Q中恰好出现在两个子句中（两者大小均为 2），即，一个子句C_u 与其在 P 中的左邻居 u 相连，另一个子句 C_w 与其在 P中的右邻居 w相连。最后，v、$\neg v \in C_u \cup C_w$，即，v 在 C_u 和 C_w中以互补形式出现。

证明。由于 P 的内部顶点中没有一个属于 $V(D)$，且 P 至少有两个内部顶点，因此 G_Q 不可能包含一个包含 P 中某条边的三角形。因此，P 的所有边都是干净边。

v 只出现在子句 C_u 和 C_w 中这一事实，是由以下事实得出的：v 在 H 中的度为二，并且我们假设 H 的每条干净边 e 都恰好对应一个子句。最后，v、$\neg v \in C_u \cup C_w$ 是因为否则 v 将是一个纯文字，但我们假设 Q 不包含纯文字。 □

引理 XII.5. 设 P 是 H 的一条干净路径，且至少有两个内部顶点。设 v 是 Q 前缀中所有 P 内部顶点里最内层的变量，并设C_u 和 C_w 是包含 v（及其邻居 u 和 w）的两个子句（见观察 XII.4）。那么，以下情形之一成立：

- v 是存在玩家，并且 Q 等价于公式$Q_\exists$；该公式由 Q 移除 C_u 和 C_w，并改为添加子句 $(C_u \cup C_w) \setminus \{v, \neg v\}$得到。
- v 是全称玩家，并且 Q 等价于公式$Q_\forall$；该公式由 Q 移除 C_u 和 C_w，并改为添加子句 $C_u \setminus \{v, \neg v\}$ 和$C_w \setminus \{v, \neg v\}$得到。

证明. 根据观察 XII.4，我们得到$C_u = l_u \vee l_v$ 和 $C_w = l_w \vee \neg l_v$，其中l_u、l_w和 l_v 分别是变量 u、w和 v上的文字；此外，C_u 和 C_w 是 Q中仅有的包含 v的子句。

我们首先证明 v为存在玩家时的引理。为证明 Q 与 $Q_\exists$ 之间等价关系的正向方向，假设Q 可满足，并且具有存在玩家的获胜策略$\mathcal{T}$。那么，对于全称玩家的每一场对局 δ，都有 $\alpha(\mathcal{T}, \delta)$必须满足 C_u 和 C_w，这反过来意味着 $\alpha(\mathcal{T}, \delta)$满足 $(C_u \cup C_v) \setminus \{v, \neg v\}$。因此，$\mathcal{T}$ 也是 $Q_\exists$ 的存在玩家获胜策略，这表明 $Q_\exists$ 可满足。现在，假设 $Q_\exists$可满足，并令 $\mathcal{T}$ 为证明这一点的 $Q_\exists$ 的存在玩家获胜策略。由于 P 至少有两个内部顶点，因此 u 或 w 必为P的内部顶点；下文不失一般性地假设 u 是 P的内部顶点。此外，由于 v 是 Q 前缀中最内层的变量（在 P的所有内部顶点中），因此可知，在 Q的前缀中，v 出现在 u 之后。令 $\mathcal{T}'$ 为从 Q 获得的策略

使得，如果针对 u的赋值不满足 C_u，则 $\mathcal{T}'$ 将 v 设置为 l_v（再次注意，由于 v 在量词前缀中位于 u之后，因此一旦 $\mathcal{T}'$ 需要决定 v的赋值，u 的赋值就已经确定）；否则将 v 设置为 $\neg l_v$。我们声称，$\mathcal{T}'$ 是 Q的存在玩家获胜策略。为证明这一点，首先注意，由于 $Q_\exists$中的子句 $(C_u \cup C_w) \setminus \{v, \neg v\}$，对于全称玩家的每一场对局 δ，$\alpha(\mathcal{T}, \delta)$必满足 l_u 或 l_w。现在，如果 $\alpha(\mathcal{T}, \delta)$满足 l_u (l_w)，那么对于 $\alpha(\mathcal{T}', \delta)$，子句 C_u 由 l_u (l_v) 满足，而子句 C_w 由 $\neg l_v$ (l_w)满足。这表明$\mathcal{T}'$ 是 Q 的存在玩家获胜策略，因此Q 可满足，如所要求。

现在继续讨论 v 为全称玩家的情况，也就是说，我们将证明：Q 不可满足，当且仅当$Q_\forall$也不可满足。设 Q 不可满足，并令 Λ 为 Q的全称获胜策略。我们声称，Λ 也是 $Q_\forall$的全称获胜策略。为证明这一点，令 δ 为存在玩家的任意一场对局。那么，$\alpha(\Lambda, \delta)$ 不满足 C 的某个子句 Q。如果 C 也属于 $Q_\exists$，那么 $Q_\forall$ 同样不被满足。否则，$C = C_u$（或 $C = C_w$），因此 $\alpha(\Lambda, \delta)$ 不满足l_u (l_w)，这表明 $\alpha(\Lambda, \delta)$ 不满足 $Q_\forall$，如需证明。现在设 $Q_\forall$ 不可满足，并令 Λ 为 $Q_\forall$的全称获胜策略。由于P 至少有两个内部顶点，因此 u 或 w 是 P 的内部顶点；下文中不失一般性地假设 u 是 P 的内部顶点。此外，由于 v 是 Q的前缀中的最内层变量，因此可知 v 在 Q的前缀中出现在 u 之后。令 Λ' 为从 Λ 获得的 Q 的策略，使得如果 u的赋值不满足 C_u，则 Λ' 将 v 设为 $\neg l_v$（再次注意，由于 v 在量词前缀中位于 u之后，因此一旦 Λ' 需要决定 v的赋值，u 的赋值就已经确定），否则 v 被设为 l_v。我们声称，Λ' 是 Q的全称获胜策略。为证明这一点，令 δ 为存在玩家在 $Q_\forall$ 上的任意一场对局（同时也是在Q上的对局）。由于 Λ' 在 $Q_\forall$上获胜，因此 $\alpha(\Lambda', \delta)$ 不满足 $Q_\forall$ 的某个子句 C。显然，如果 C 也属于 Q，则无需再证明其他内容。否则，$C = l_u$（或 $C = l_w$），因此 $\alpha(\Lambda', \delta)$ 不满足C_u（或 C_v），如需证明。因此，Λ' 是 Q 的全称获胜策略，从而 Q 不可满足，如需证明。

□

注意，在穷尽地应用引理 XII.5 后，我们得到一个实例，其中 H 中的每条干净路径至多有 2 个内部顶点。现在我们可以证明定理 V.2 了。

定理 V.2 的证明。令 Q 为对原始公式穷尽地应用上述所有归约规则后得到的实例。此外，令 G 为 Q 的原始图，D 为 G 的一个最小反馈边集，$k = |D|$，并令 $H = G - D$。那么，H 中的每条干净路径至多有 2 个内部顶点。此外，H 中（按包含关系）极大干净路径的数量至多为 $2k+2k-2-1 = 4k-3$。这是因为，如上所述，H 至多有 $2k$ 个叶节点，且至多有 $2k-2$ 个度大于 2 的顶点，而任何含有 $2k+2k-2$ 个顶点的树至多有 $2k+2k-3$ 条边。因此，H 的顶点数（边数）至多为 $2k+2k-2+2$（$4k-3$）$= 12k-8$（$12k-9$）。由此可知，Q 的变量数至多为 $12k-8$；接下来只需界定 Q 的子句数。由于 H 中端点不在 D 内的每条边都是干净边，因此 H 中除至多 $2k$ 条边外，其余边都是干净边，因而每条至多为 Q 贡献一个子句。此外，在最坏情况下（就可得到的子句数而言），H 中剩余的 $2k$ 条边与 $|D|$ 中的边可以构成一个大小至多为 $\lfloor (\sqrt{24k+1}+1)/2 \rfloor$ 的团（这是因为，含有 $2k+k = 3k$ 条边的团的顶点数 n 满足 $\binom{n}{2} = 3k$）。因此，核中来自 H 的干净边的子句至多有 $12k-9-2k = 10k-9$ 个，来自 G 的剩余边的子句至多有 $3^{\lfloor(\sqrt{24k+1}+1)/2\rfloor}$ 个，所以 Q 的子句总数至多为 $10k-9+3^{\lfloor(\sqrt{24k+1}+1)/2\rfloor}$。

注意，如果我们将范围限制在 c-合取范式量化布尔公式，那么得到的核中团的大小至多为 c。因此，G 中不干净的 $3k$ 边至多能贡献 $3k/\binom{c}{2}$ 个团，且每个团的大小至多为 c，这使得由这些边产生的子句数量减少到 3^c $(3k/\binom{c}{2})$，该数量关于 k 呈线性增长。 □

定理 V.5 的证明。设 Q 是给定的合取范式量化布尔公式，设 $D \subseteq E(I_Q)$ 是 I_Q 的一个反馈边集，即下文记作 F 的图 $I_Q - D$ 是一片森林。与原始图的情形一样，我们首先注意到，可以假设 Q 不包含任何单位子句或纯文字。考虑一个不属于 $V(D)$ 的叶节点 l。那么，l 不可能是子句，因为 Q 不包含单位子句；l 也不可能是变量，因为 Q 不包含纯文字。因此，F 的每个叶节点都属于$V(D)$，这意味着 F 至多有 $2k$ 个叶节点。由于这意味着 F 至多有 $2k-2$ 个度大于 2 的顶点，因此只需界定 F 中度恰为 2 的顶点数量。与原始图的情形一样，如果 P 是 F 的一条路径，并且 P 的所有内部顶点在 F 中的度均为二，且不属于 $V(D)$，我们就称 P 是一条干净路径。设 P 是一条干净路径，且 v 是一个变量，同时也是 P 的内部顶点，并且 v 的两个邻居也都是 P 的内部顶点。那么，v 满足观察 XII.4 所蕴含的所有性质，即 v 恰好包含在两个子句 C_u 和 C_w 中，且这两个子句的元数均为 2（即它们是 P 上的邻居），并且 v 在 C_u和 C_w 中以互补形式出现。因此，对于包含这样一个

变量 v 的任意干净路径，我们都可以应用引理 XII.5 消去 P 上至少一个变量。由于当 P 至少有 5 个内部顶点时，这样的变量总是存在，因此可以假设每条干净路径至多有 5 个内部顶点。使用与原始图情形相同的论证，我们得到 F 至多包含 $4k-3$ 条（按包含关系计）极大干净路径。因此，F 至多有 $5(4k-3)+2k+2k-2=24k-17$ 个顶点，从而约简公式 Q 至多有同样数量的变量和子句。 □

XIII. 第 VI 节省略的证明（探索原始图上 CQSAT 可处理性的极限）

命题 VI.1. 设 $\mathcal{P}$ 是任意可高效计算的性质，且 Q 是一个合取范式量化布尔公式。那么，计算满足 $\mathcal{P}(Q,D,c)$ 的 Q 的最小 c-删除集 D 是以 $|D|+c$ 为参数的固定参数可处理问题。

证明。 [21, 定理 12] 表明，判定一个图 G 是否具有大小至多为 k 的 c-删除集，是以 $k+c$ 为参数的固定参数可处理问题。该证明使用了一种有界深度搜索树算法；实际上，该算法可以在以 $k+c$ 为参数的固定参数可处理时间内，枚举所有大小至多为 k 的可能 c-删除集。因此，利用该算法，我们可以在所需时间内枚举合取范式量化布尔公式 Q 的所有大小至多为 k 的 c-删除集。然后，对于每个这样的 c-删除集 D，我们可以使用用于判定性质 $\mathcal{P}$ 的算法，判断 $\mathcal{P}(Q,D,c)$ 为真还是为假。接着，我们可以返回使 $\mathcal{P}(Q,D,c)$ 为真的最小集合；如果不存在这样的集合，则返回 false。最后，从 $k=1$ 开始，只要算法返回 false，就将 k 递增，我们便可以找到 Q 的最小 c-删除集。 □

我们需要以下概念。设 Q 为一个合取范式量化布尔公式，c 为一个整数，且 $D\subseteq \mathrm{var}(Q)$ 是 c-删除集。我们定义 G_F-D的分量集合上的如下等价关系 $\sim$。也就是说，若 G_F-D 中的两个分量 C 和 C' 满足 $C\sim C'$，且存在一个双射 $\eta:\mathrm{var}(C)\to\mathrm{var}(C')$，使得 $C'\cup D(Q)$ 等于将 C 中的所有变量按照 η 重命名（为 C'中的变量）后得到的公式 $C\cup D(Q)$，则二者等价。因此，对于 G_Q 的子图或顶点集合，记 $Q(A)$ 为从 Q中删除 A 之外的所有变量，以及至少含有一个 A之外变量的所有子句后得到的合取范式量化布尔公式。我们称满足 $C\sim C'$ 的两个分量 C 和 C' 具有相同的（分量）类型，并用 $\mathcal{T}(Q,D)$ 表示 Q 的所有分量类型的集合（相对于 D）。此外，对于类型 $t\in\mathcal{T}(Q,D)$，我们用 $\mathcal{C}(Q,D,t)$ 表示 Q 中所有类型为 t 的分量所构成的集合。

命题 XIII. 1 $\sim$ $2|D|+$。该关系至多有 $1\ c2^{3^{c+|D|}}$ ( ) 个等价类。

证明。设 C 是 G_Q-D 的一个分量。那么，C 的类型完全由以下信息刻画：相对于 D 中的变量的位置，以及 C 中的变量以何种方式（存在玩家或全称玩家）出现在 Q 的前缀中；此外还包括包含 C 中变量的子句集合。由于 C 的每个变量 v 在 Q 的前缀中至多可放置于 $|D|+1$ 个不同位置，并且只能进行存在量化或全称量化，因此 C 中的变量在 Q 的前缀中出现的不同方式至多有 $(2|D|+1)^c$ 种。此外，由于 C 中的变量只能与 $C\cup D$ 中的变量共同出现在一个子句中，因此包含 C 中变量的不同子句至多有 $3^{c+|D|}$ 个，从而这样的子句至多形成 $2^{3^{c+|D|}}$ 个不同的集合。因此，分量的不同类型至多有 $(2|D|+1)^c2^{3^{c+|D|}}$ 种，证毕。 □

命题 XIII.2. 设 Q 是一个合取范式量化布尔公式，且 D 是 G_Q的一个c-删除集。那么，对于每个 $t\in\mathcal{T}(Q,D)$，我们都可以在多项式时间内计算出 $\mathcal{T}(Q,D)$ 和 $\mathcal{C}(Q,D,t)$。

证明。设 $C_1,\ldots,C_m$是 G_Q-D的分量。我们先令 $T=\emptyset$，然后对每个满足$1\le i\le m$的 i，执行以下操作：如果 $T=\emptyset$，则将集合 $\{C_i\}$ 加入 T。否则，我们遍历每个 $t\in T$，并按如下方式检验 C_i 是否属于类型 t。任取 $C_t\in t$。如果 $\mathrm{var}(C_i)|\neq|\mathrm{var}(C_t)|$ 满足 $|$，则 C_i 不属于类型 t，我们继续检查下一个 $t\in T$。否则，$\eta:\mathrm{var}(C_i)\to\mathrm{var}(C_t)$ 是唯一的映射，它将 $Q(C_i)$ 的前缀中的第 i 个变量映射到 $Q(C_t)$ 的前缀中的第 i 个变量。接着，我们检查 $Q(C_t\cup D)$ 是否等于将 $Q(C_i\cup D)$ 中的每个变量按照 η 重命名后得到的公式。注意，只需遍历 $Q(C_t\cup D)$ 的所有变量和子句，即可在多项式时间内完成此操作。如果相等，则 C_i 属于类型 t，我们将 C_i 加入 t；否则，继续检查下一个 $t\in T$。最后，如果不存在任何满足 C_i 属于类型 t的 $t\in T$，则将集合 $\{C_i\}$ 加入 T。以这种方式检查完所有分量后，所得集合 T 等于 $\mathcal{T}(Q,D)$，并且对于每个 $t\in T$，集合 $\mathcal{C}(Q,D,t)$ 等于 t。 □

现在我们证明，可以消除合取范式量化布尔公式的 c- 删除集 D 中的所有全称变量，同时不会破坏公式的结构；也就是说，在消除所有全称变量后，我们得到一个公式 Q' 以及一个大小至多为 2^cc的 2^cc-删除集。

命题 VI.2. 设 Q 是一个合取范式量化布尔公式，且 D 是 Q 的 c- 删除集。那么，在 $\mathcal{O}(2^u\|Q\|)$ 时间内，其中 $u=|D\cap\mathrm{var}^{\forall}(Q)|$，我们可以构造一个等价的合取范式量化布尔公式$Q'$ 以及一个集合 $D'\subseteq\mathrm{var}^{\exists}(Q')$，满足 $|D'|\le 2^u|D|$，使得 D' 是 Q' 的 2^uc-删除集。

证明。设 $Q = Q_1 v_1 Q_2 v_2 \cdots Q_n v_n F$。如果 $\mathrm{var}^{\forall}(D) = \emptyset$，那么我们直接返回 $Q' = Q$ 和 $D' = D$。否则，令 $v_i \in \mathrm{var}^{\forall}(D)$ 为 D 中的全称变量，该变量是 Q 前缀中的最内层变量。我们将使用量词展开来消除 v_i。也就是说，令 $Q(v_i)$ 为从 Q 中消除 v_i 后得到的公式，即：

$$Q(v_i) = Q_1 v_1 \cdots Q_{i-1} v_{i-1}((Q_{i+1} v_{i+1} \cdots Q_n v_n) F[v_i = 0]) \wedge (Q_{i+1} v_{i+1} \cdots Q_n v_n F[v_i = 1]))$$

注意，$Q(v_i)$与 Q 等价，并且可以在 $\mathcal{O}(\|Q\|)$时间内计算得到。然而，$Q(v_i)$并非前束范式。为了将 Q_{v_i} 转换为前束范式，我们为每个变量 $v \in \{v_{i+1}, \ldots, v_n\}$ 引入 v' 个副本，然后将 $Q(v_i)$重写为等价公式 $Q_c(v_i)$，如下所示：

$$Q_c(v_i) = Q_1 v_1 \cdots Q_{i-1} v_{i-1} Q_{i+1} v_{i+1} \cdots Q_n v_n Q_{i+1} v'_{i+1} \cdots Q_n v'_n F[v_i = 0]) \wedge F'[v_i = 1]$$

其中，F' 是通过将 F 中的变量 $v \in \{v_{i+1}, \ldots, v_n\}$ 的每次出现重命名为 v'后得到的合取范式公式。于是：

- $Q_c(v_i)$的大小至多是 Q 的两倍，并且可以在 $\mathcal{O}(\|Q\|)$ 时间内计算得到，
- 令 A 为 D中所有（存在）变量的集合，这些变量在 Q 的前缀中出现在 v_i 之后，并令 $A' = \{v' \mid v \in A\}$。那么，$D'' = (D \setminus \{v_i\}) \cup A'$ 是 Q'_{v_i}的一个 $2c$-删除集。此外，D'' 比 D少一个全称变量。因此，$Q_c(v_i)$具有一个$2c$-删除集，其大小至多为 D 的两倍，并且少一个全称变量。

因此，如果我们按照 Q 的前缀中出现的逆序，对 D 中的每个全称变量重复上述过程，就会得到一个等价公式 Q'，它具有一个大小至多为 $2^u|D|$ 的 $2^u c$-删除集 D'，并且 $D' \subseteq \mathrm{var}^{\exists}(Q')$ 的计算时间为 $\mathcal{O}(\|Q'\|) = \mathcal{O}(2^u \|Q\|)$。 □

A. $\exists^{\leq 1}\forall^u$ 类型分量的算法

引理 XIII.3。设 Q 是一个合取范式量化布尔公式，且 $D \subseteq \mathrm{var}(Q)$ 是 Q 的一个 c-删除集。设 $t \in \mathcal{T}(Q)$ 具有 $\exists^{\leq 1}\forall$ 的形式，其中 $\mathcal{C}(Q, D, t) = \{A_1, \ldots, A_r\}$，并且 $(A_1, \ldots, A_r)$ 是按照各分量中唯一存在变量的出现顺序排列的 t 类型分量。则 Q 等价于公式 Q'：该公式由 Q 删除包含分量 $\{A_{2^c}, \ldots, A_r\}$ 中变量的所有子句（以及分量 $\{A_{2^c}, \ldots, A_r\}$ 中的所有变量）后得到。

证明。我们将证明引理适用于 t 具有 $\exists\forall^u$ 形式的情形，这是最一般的情形，其中 u 是 t 类型分量中的全称变量数量。令 x_i 为 A_i 的唯一存在变量，并令 Y_i 为 A_i 中所有全称变量的集合。注意，$x_1, \ldots, x_r$ 是存在变量出现的顺序在 Q 的前缀中。此外，令 $X = \{x_1, \ldots, x_n\}$、$X_P = \{x_1, \ldots, x_{2^{u+1}-1}\}$，并令 $Y = \bigcup_{i=1}^{r} Y_i$。

我们通过证明 Q 有存在玩家的获胜策略，当且仅当 Q' 有，来证明该引理。命题的正向显然成立，因为 Q' 的子句是 Q 的子句的子集。为证明反向，设 $\mathcal{T}' = (\tau'_x \colon \{0,1\}^{\mathrm{var}(Q')^{\forall}_{<x}} \to \{0,1\})_{x \in \mathrm{var}^{\exists}(Q')}$ 是 Q' 的一个存在玩家获胜策略。我们将说明如何构造 $\mathcal{T} = (\tau_x \colon \{0,1\}^{\mathrm{var}(Q)^{\forall}_{<x}} \to \{0,1\})_{x \in \mathrm{var}^{\exists}(Q)}$，使其成为 Q 的一个存在玩家获胜策略。

令 $x \in \mathrm{var}^{\exists}(Q)$，并令 $\delta \colon V^{\forall}_{<x} \to \{0,1\}$ 是 Q 的前缀中出现在 x 之前的全称变量所构成的任意一局对局。令 δ' 为 δ 在 Q' 的变量上的限制，即 $\delta' = \delta_{\mathrm{var}(Q')}$。现在，我们按如下方式迭代地定义变量 $\mathrm{var}(\delta')$ 上的一个更强对局 δ^S。令 $(y_1, \ldots, y_n)$ 为 $\mathrm{var}(\delta')$ 中按照其在 Q' 的量词前缀中出现顺序排列的变量。我们设定 $\delta^S = \delta^S_n$ 和 $\delta^S_0 = \delta'$。此外，对于每个满足 $0 < i \leq n$ 的 i，δ^S_i 按如下方式从 δ^S_{i-1} 得到。

- 如果 y_i 不是类型为 t 的分量中的变量，即$y_i \notin Y$，那么我们令 $\delta^S_i = \delta^S_{i-1}$。
- 否则，y_i 是某个分量 A_l 的第 a 个全称量化变量，其中 $1 \leq a \leq u$且 $1 \leq l \leq 2^{u+1} - 1$。我们首先令$\delta^S_i = \delta^S_{i-1}$。设 β：$X_P \cap \mathrm{var}(\delta^S_{i-1})$是在 $X_P \cap \mathrm{var}(\delta^S_{i-1})$中存在变量的赋值，该赋值是在 $\mathcal{T}'$ 与 δ^S_{i-1}进行对局时得到的，即 $\beta(v) = (\alpha(\mathcal{T}', \delta^S_{i-1}))_{X_P \cap \mathrm{var}(\delta^S_{i-1})}$。注意到，在 Q' 的前缀中，x_l 出现在 y_i 之前，因此$\beta(x_j)$是良定义的。设 S 为所有索引 j的集合，使得 $\delta^S_{i-1}(Y^a_j) = \delta^S_{i-1}(Y^a_l)$ 且 $\beta(x_l) = (\alpha(\mathcal{T}', \delta^S_{i-1}))(x_j)$，即分量 $(A_1, \ldots, A_{2^{u+1}-1})$ 的所有索引：当 $\mathcal{T}'$ 与 δ^S_{i-1}进行对局时，唯一存在变量的赋值以及前 a 个全称变量的赋值都与 A_l一致。于是，如果 $|S| \leq 2^{u-a}$ 且$\delta^S_i(y_i) = 1 - \delta^S_{i-1}(y_i)$，我们令 $\delta^S_i(y_i) = \delta^S_{i-1}(y_i)$；否则，

现在我们可以如下定义 $\tau_x(\delta)$。

- 若 $x \in \mathrm{var}^{\exists}(Q')$，则令 $\tau_x(\delta) = \tau'_x(\delta^S)$
- 否则，对于某个 i，满足 $2^{u+1} \leq i \leq r$。因此，x 出现在所有变量x_1、……、$x_{2^{u+1}-1}$之后，并且每个这样的变量 x_j都已被赋值为 $\tau'_{x_j}(\delta^S)$ $b \in \{0,1\}$。令为 $\tau'_{x_1}(\delta^S)$…… $\tau'_{x_{2^{u+1}-1}}(\delta^S)$这些值中出现次数最多的值。注意，$b$至少出现 $(2^{u+1} - 1)/2 \geq 2^u$ 次。然后，令 $\tau_x(\delta) = b$。

这就完成了对 $\mathcal{T}$的定义，接下来还需证明 $\mathcal{T}$ 确实是存在玩家在 Q上的获胜策略。为此，我们首先证明

从 δ得到的赋值 δ^S 具有以下性质。

否则，则存在一个全称方博弈 δ： $V^\forall \to \{0,1\}$，使得 $\alpha(\mathcal{T},\delta)$ 不满足 Q 的某个子句 C。

如果 C 不在某个分量 A_i 中，那么 C 也属于 Q，而且 C 中的所有变量都按照 $\alpha(\mathcal{T}',\delta^S)$ 进行赋值。因此， $\alpha(\mathcal{T}',\delta^S)$ 将不满足 Q'，这与我们的假设相矛盾： $\mathcal{T}'$ 是 Q' 的获胜存在策略。因此，我们可以假设 C 位于某个分量 A_i 中。

如果 $i \geq 2^{u+1}$，那么 $\tau_{x_i}(\delta)$ 取值为 b，该值在 $\tau'_{x_1}(\delta^S), \ldots, \tau'_{x_{2^{u+1}-1}}(\delta^S)$ 中至少出现 2^u 次。因此，根据 δ^S 的定义，存在一个 $1 \leq j \leq 2^{u+1}-1$，使得 $\tau_{x_j}(\delta) = b$ 且 $\delta^S(Y_j) = \delta(Y_i)$，这意味着 A_j 中 C 的副本 C' 也未被满足。但此时， C' 也不会被 $\alpha(\mathcal{T}',\delta^S)$ 满足，这与我们的假设相矛盾： $\mathcal{T}'$ 是 Q' 的获胜策略。

另一方面，如果 $i < 2^{u+1}$，那么根据 δ^S 的定义，存在一个 $1 \leq j \leq 2^{u+1}-1$ 使得 $\tau_{x_i}\ \delta \quad \delta^S\ Y_j\ = \delta\ Y_i \qquad \tau_{x_j} \quad =$ $(\delta)\ (\ )$ 且 $(\ )\ (\ )$，这意味着 A_j 中的 C 的副本C' 也不满足。但随后， C' 也不会被 $\alpha(\mathcal{T}',\delta^S)$ 满足，这与我们的假设相矛盾： $\mathcal{T}'$ 是 Q' 的获胜策略。 □

定理 VI.4. 设 Q 是一个合取范式量化布尔公式，并设 $D \subseteq \mathrm{var}(Q)$ 是一个 c-删除集，可将 Q 分解为 $\exists^{\leq 1}\forall^u$形式的分量。那么，判定 Q 是否可满足是固定参数可处理的，其参数为 $|D|+c$。

证明。对于每一种类型 $t \in \mathcal{T}(Q,D)$，我们使用引理 XIII.3 将类型为 t 的分量数量减少到至多 2^c。设 Q' 是对 Q 穷尽应用引理 XIII.3 后得到的公式。根据命题 XIII.1，在 $\mathcal{T}(Q,D)$ 中至多存在 $a = (2|D|+1)^c 2^{3^{c+|D|}}$种类型。因此， Q'至多包含 $a2^c c + |D| = (2|D|+1)^c 2^{3^{c+|D|}} 2^c c + |D|$个变量，从而可以通过蛮力搜索在以 $|D|+c$为参数的固定参数可处理时间内求解。 □

B. 单变量删除集

现在我们考虑由单个变量 e组成的删除集。我们可以假设变量 e是存在量化的——如果它是全称量化的，我们可以应用香农展开，得到一个可分解为变量不相交子公式的量化布尔公式，并分别对这些子公式求值。此外，不失一般性，我们可以假设 e 是量词前缀中的最内层变量，因为如果一个最内层变量仅出现在某个分量中，那么当它是全称量化时，可以通过全称约简将其移除；当它是存在量化时，则可以通过香农展开（在此情况下也称为变量消除）将其移除。请注意，对删除变量 e应用变量消除并无帮助，因为它会将来自不同分量的每对子句析取起来，并完全掩盖公式的结构来自不同的分量，并完全掩盖了公式的结构。

在本节余下部分，设 $Q = \mathrm{P}.F$ 是一个量词前缀为 $\mathrm{P} = Q_1 v_1 \ldots Q_\ell v_\ell \exists e$、矩阵为 $F_1 \wedge \cdots \wedge F_m$ 的量化布尔公式，且对于 $1 \leq i < j \leq m$， $\mathrm{var}(F_i) \cap \mathrm{var}(F_j) \subseteq \{e\}$。我们将说明，可以根据各个分量的性质高效计算 Q 的真值。对于 $1 \leq i \leq m$，我们用 $Q^{(i)} = \mathrm{P}.F_i$ 表示一个分量量化布尔公式，它与 Q 具有相同的量词前缀，矩阵为 F_i。

对于 $0 \leq i \leq n$，令 $V_i = \{v_1, \ldots, v_i\}$ 表示前缀中最前面的 i 个变量构成的集合（定义 $V_0 = \emptyset$），并令 $V_i^q = V_i \cap \mathrm{var}^q(Q)$ 表示其对量词 $q \in \{\exists, \forall\}$ 的限制。下文中，我们考虑部分全称（存在）策略，即只在 V_i 中的全称（存在）变量上定义的策略，其中 $0 \leq i \leq \ell$。若部分全称（分别为存在）策略 $\mathcal{T}$ 在布尔公式 F 中禁止（允许）赋值 σ，则任何与 $\mathcal{T}$ 和 σ 一致的赋值 β 都会使 F 为假（真）。这里，若赋值 β 扩展了 σ，且对于由 $\mathcal{T}$ 定义的每个变量 v_i，只要 β 属于 $\mathrm{dom}(\beta)$，就有 $\tau_{v_i}(\beta|_{V_i^q}) = \beta(v)$，则称赋值 β 与部分策略 $\mathcal{T}$ 及赋值 σ 一致；若 $\mathcal{T}$ 是部分全称策略，则 $q = \exists$，若 $\mathcal{T}$ 是部分存在策略，则 $q = \forall$。若策略禁止（允许）集合 Σ 中的每个赋值 $\sigma \in \Sigma$，则称其在 F 中禁止（允许）赋值集合 Σ。为简便起见，我们将文字 e 与赋值 σ： $e \mapsto 1$ 等同，将文字 $\neg e$ 与赋值 σ'： $e \mapsto 0$ 等同。

显然，当且仅当某个全称策略禁止 e 和 $\neg e$ 在 F 中出现时，它才是 Q 的获胜策略。 Q 不可满足的一个简单充分条件是：存在一个针对单个分量量化布尔公式 $Q^{(i)}$ 的全称策略，该策略已经禁止 e 和 $\neg e$。下面的引理给出了一个略弱的必要条件。

引理 XIII.4. 如果 Q 不可满足，则必定存在一个索引 $1 \leq i \leq m$ 和一个全称策略，使得对于每个赋值 $\sigma \in \{e, \neg e\}$，该策略都禁止 σ 在 F_i 中出现。

证明。如果 Q 不可满足，那么 $Q[e]$和 $Q[\neg e]$都必须不可满足。对 e 赋值后，矩阵会断开，因此对于每个赋值 $\sigma \in \{e, \neg e\}$，必定存在一个分量量化布尔公式 $Q^{(i)}$，使得 $Q^{(i)}[\sigma]$不可满足。相应的全称获胜策略会禁止赋值 σ 在 F_i 中出现。 □

如果存在不同的分量量化布尔公式，分别禁止 e 和 $\neg e$，那么相应的策略可以合并为一个同时实现两者的策略。这是因为除了删除变量 e 之外，这些分量量化布尔公式没有共享变量。更一般地，由某个分量（或一组分量）中的子句所决定的任何策略性质

在仅更改那些未出现在该分量（或该组分量）中的变量的策略函数时，都会得以保留。

定义 XIII.5. 若在变量 V_i 上定义的（部分）策略类 $\mathcal{C}$ 满足：对于变量 w，只要 $\mathcal{T} \in \mathcal{C}$ 且 $\mathcal{T}'$ 是一个仅在 $w \in V_i$ 上与 $\mathcal{T}$ 不同的部分策略，就有 $\mathcal{T}' \in \mathcal{C}$，则称 $\mathcal{C}$ 对变量 w 不敏感。若 $\mathcal{C}$ 对变量集合 W 中的每个变量 $w \in W$ 都不敏感，则称其对集合 W 不敏感。

引理 XIII.6. 设 $\mathcal{C}_1$ 是在变量 V_i 上定义的策略（部分策略）类，对 W_1 不敏感；设 $\mathcal{C}_2$ 是在 V_i 上定义的部分策略类，对 W_2 不敏感，且满足 $V_i \subseteq W_1 \cup W_2$。则 $\mathcal{C}_1$ 与 $\mathcal{C}_2$ 的交集非空。

证明。设 $\mathcal{T}_1 \in \mathcal{C}_1$ 且 $\mathcal{T}_2 \in \mathcal{C}_2$。考虑策略 $\mathcal{T} = \{f_v \in \mathcal{T}_1 \mid v \in V_i \setminus W_1\} \cup \{f_w \in \mathcal{T}_2 \mid w \in V_i \cap W_1\}$。由于它仅在变量 W_1 中的变量上与 $\mathcal{T}_1$ 不同，且 $\mathcal{T}_1$ 对 W_1 不敏感，因此有 $\mathcal{T} \in \mathcal{C}_1$。类似地，$\mathcal{T}$ 仅在变量 $V_i \setminus W_1 \subseteq W_2$ 中的变量上与 $\mathcal{T}_2$ 不同，因此 $\mathcal{T} \in \mathcal{C}_2$ 同样成立。 □

这一结果可以应用于禁止或允许对变量 e进行某些赋值的策略。

引理 XIII.7. 设 F 是一个布尔公式，且 var(F) ⊆var(Q)，并且 W =var(Q) \var(F)。以下部分策略类与W无关：

1）禁止在 F中$\Sigma \subseteq \{e, \neg e\}$ 的部分全称策略类。2）允许在 F中$\Sigma \subseteq \{e, \neg e\}$ 的部分存在策略类。

证明。部分策略 $\mathcal{T}$是否禁止或允许对 F 的赋值，仅取决于与 $\mathcal{T}$一致的 β是否满足或不满足 F，而改变 F之外变量的函数不会影响这一点。 □

引理 XIII.8. 设 F和 F' 是布尔公式，且 var(F)∩var(F') ⊆ {e}，并且 var(F)∪var(F') ⊆ Q。设 $\mathcal{T}$ 是一个在 F 中禁止 e 的全称策略，$\mathcal{T}'$是一个在 F'中禁止 $\neg e$的全称策略。那么 P.$F \wedge F'$具有一个全称获胜策略。

证明。根据引理 XIII.7，部分策略 $\mathcal{T}$ 与 var(Q) \ var(F)无关，并且 $\mathcal{T}'$ 与 var(Q)\var(F')无关。由于 (var(Q)\var(F))∪(var(Q)\var(F')) = var(Q)，根据引理 XIII.6，必定存在一种全称策略$\mathcal{T}''$，在 F中禁止 e，并在 F' 中禁止 $\neg e$。因此，$\mathcal{T}''$ 在 $F \wedge F'$中同时禁止 e 和 $\neg e$，并且它是一种获胜的通用策略。 □

我们可以得出结论：只要存在不同的分量量化布尔公式，以及禁止 e 和 $\neg e$ 的全称策略，就存在一种全称获胜策略。

下一个引理表明，任何分量 F_i，只要存在一种（部分）存在策略，允许在 F_i 中同时出现 e 和 $\neg e$，就可以从矩阵中移除，而不改变真值。

引理 XIII.9。设 F和 F' 是满足以下条件的布尔公式：var(F)∪var(F') ⊆var(Q)，且 var(F)∩var(F') ⊆{e}。设 $1 \leq i \leq \ell$，并设 $\mathcal{T}$ 是在 $V_i^{\exists}$ 上定义的部分存在策略，且 $\mathcal{T}$允许 e 和¬e出现在 F'中。那么，P.F和 P.$F \wedge F'$ 具有相同的真值。

证明。如果 P.F为假，则结论显然成立。否则，存在一个针对 P.F的存在玩家的获胜策略 $\mathcal{T}'$。令 $f_e \in \mathcal{T}'$ 为变量 e的策略函数，并令 $\mathcal{T}''$ 为从 $\mathcal{T}'$中移除 f_e 后得到的部分存在策略。根据构造，通过添加函数 f_e，该部分策略可以转化为针对 P.F 的存在玩家的获胜策略，并且相应的部分存在策略类与 $V_i^{\exists} \setminus$ var(F)无关。根据引理 XIII.7，允许在 F' 中使用 e 和¬e 的部分存在策略类与 $V_i^{\exists} \setminus$ var(F')无关。此外，($V_i^{\exists} \setminus$ var(F)) ∪ ($V_i^{\exists} \setminus$ var(F')) = $V_i^{\exists}$，因此我们可以应用引理 XIII.6，并得出结论：存在一个定义在 $V_i^{\exists}$上的部分存在策略，该策略允许在 F' 中使用 e 和 $\neg e$，并且通过用 f_e扩展该策略，可以将其转化为针对 P.F 的存在玩家的获胜策略。通过将 f_e添加到该部分策略中，我们得到针对 P.$F \wedge F'$的存在玩家的获胜策略。 □

我们简要总结一下。现在可以假设，不存在这样的分量：其中存在一种部分全称策略，禁止 e 的两种赋值（在此情况下 Q 为假）；或者存在一种部分存在策略，允许两种赋值（这些策略不会影响真值）。此外，我们还可以假设，不存在不同的分量，使得全称玩家分别能够禁止 e 和 $\neg e$（在此情况下 Q 为假）。根据引理 XIII.4，要使 Q 为假，必须存在一个分量 F_i，使得其中存在一种全称策略，在 F_i 中禁止 e，并存在另一种全称策略，在 F_i 中禁止 $\neg e$。然而，我们的假设表明，不存在能够在 F_i 中禁止两种赋值的策略。对于 $j \neq i$ 而言，任何剩余的分量 F_j 都必须具有对偶性质：存在一种部分存在策略，在 F_j 中允许 e，并存在另一种部分存在策略，在 F_j 中允许 $\neg e$，但不存在能够允许两种赋值的策略。

因此，对于每个剩余的分量，在任何对局中的某个时刻，全称玩家（分别地，存在玩家）必须决定禁止（允许） e 或 $\neg e$。在此之前，存在一种能够让这两个选项继续保留的策略。

定义 XIII.10。设 F 是一个布尔公式，且 var(F) ⊆ var(Q)，设 $0 \leq i \leq \ell$，并设 $\mathcal{T}$ 是定义在

$V_i^\forall$($V_i^\exists$) 上的部分全称玩家（分别为存在玩家）策略，满足以下性质：对于任意赋值 β： $V_i \to \{0,1\}$，只要其与 $\mathcal{T}$ 一致，全称玩家（存在玩家）就在 $V_\ell^\forall \setminus V_i$（$V_\ell^\exists \setminus V_i$）上定义了一个部分策略，该策略在 $F[\beta]$ 中禁止（允许） e，并且还有一个在 $F[\beta]$ 中禁止（允许） $\neg e$ 的策略。我们称 $\mathcal{T}$ 在索引 i（在 F 中）留下了一个选择。

存在一种部分策略的对偶概念，它确保在某个时刻，全称玩家可以禁止某个赋值（相应地，存在玩家可以允许某个赋值）。

定义 XIII.11. 设 F 是一个布尔公式，且 var(F) $\subseteq$ var(Q)，设 $0 \le i \le \ell$，并设 $\mathcal{T}$ 是定义在 $V_i^\forall$($V_i^\exists$) 上、满足以下性质的部分全称（相应地，存在）策略：对于任意与 $\mathcal{T}$ 一致的对局 β： $V_i \to \{0,1\}$，存在一个赋值 $\sigma_\beta \in \{e, \neg e\}$，使得全称（存在）玩家拥有一个定义在 $V_\ell^\forall \setminus V_i$ ($V_\ell^\exists \setminus V_i$) 上的部分策略，该策略在 $F[\beta]$ 中禁止（允许） σ_β。此时，我们说 $\mathcal{T}$ 在索引 i（在 F 中）赋能。

引理 XIII.12. 设 F 是一个布尔公式，且 var(F) $\subseteq$ var(Q)，并设 $0 \le i \le \ell$。在 i 处留下一个选择的部分全称（相应地，存在）策略类，以及在 i 处赋能的部分全称（相应地，存在）策略类，都与 $V_i \setminus$ var(F) 无关。

证明。对于每个与 $\mathcal{T}$ 一致的对局 β，部分策略 $\mathcal{T}$ 是留下一个选择还是在 i 处赋予权限，仅取决于 $F[\beta]$；而修改 β 中不出现在 F 里的变量 V_i，不会改变此公式。□

引理 XIII.13。设 F 和 F' 是布尔公式，其中 var(F) $\cap$ var(F') $\subseteq \{e\}$，且 var(F) $\cup$ var(F') $\subseteq$ var(Q)。如果存在一个在 i 处赋予权限的部分全称玩家（分别为存在玩家）策略 $\mathcal{T}$，其作用于 F；并且存在一个在 i 处留下选择的部分全称玩家（存在玩家）策略 $\mathcal{T}'$，其作用于 F'，那么对于 P.$F \wedge F'$，存在一个全称玩家（存在玩家）的获胜策略。

证明。根据引理 XIII.12 和引理 XIII.6，存在一个针对 $V_i^\forall$（$V_i^\exists$）的部分全称玩家（存在玩家）策略 $\mathcal{T}$，使得对于与 $\mathcal{T}$ 一致的任意对局 β：全称玩家（存在玩家）拥有一个部分策略，禁止（允许）在 $F[\beta]$中出现 $\sigma_\beta \in \{e, \neg e\}$，以及一个部分策略，禁止（允许）在 $F'[\beta]$中出现 $\neg\sigma_\beta$（分别为 σ_β）。根据引理 XIII.7 和引理 XIII.6，这意味着存在一个部分全称玩家（存在玩家）策略 $\mathcal{T}_\beta$，禁止（允许）在 F 中出现 σ_β，并禁止（允许）在 F'中出现 $\neg\sigma_\beta$（σ_β）。也就是说，与 β 和 $\mathcal{T}_\beta$一致的任意对局都会使 $F \wedge F'$为假（对于存在玩家策略的情况，与 β、$\mathcal{T}_\beta$和 σ_β一致的任意对局都会使 $F \wedge F'$为真）。将 $\mathcal{T}$与 $\mathcal{T}_\beta$ 结合（对于存在玩家策略，还结合根据 σ_β为 e 赋值的函数），即可得到 P.$F \wedge F'$的全称玩家（存在玩家）获胜策略。□

“赋能”和“保留选择”这两个概念具有如下对偶关系：当且仅当另一位玩家不存在在 i处“保留选择”的策略时，一位玩家才拥有在 i 处“赋能”的策略。

引理 XIII.14。设 F 是一个满足 var(F) $\subseteq$ var(Q) 的布尔公式，并设 $0 \le i \le \ell$。当且仅当不存在在 i处赋予权限的部分存在玩家（全称玩家）策略时，才存在一个在索引 i 处留下选择的部分全称玩家（分别为存在玩家）策略。

证明。考虑全称玩家与存在玩家之间的赋值博弈，其中当且仅当存在赋值 $\sigma_\beta \in \{e, \neg e\}$ 和一个部分全称策略，能够在 $F[\beta]$ 中禁止 σ_β 时，全称玩家在对局 β： $V_i \to \{0,1\}$ 中获胜。该博弈是确定的，因此，要么存在一个部分全称策略，能够至少在 i 处赋予（如果全称玩家拥有获胜策略），要么存在一个存在策略 $\mathcal{T}$，使得对于任何与 $\mathcal{T}$ 一致的 β，都不存在 $\sigma \in \{e, \neg e\}$，使全称玩家拥有一个策略，能够在 $F[\beta]$ 中禁止 σ（如果存在玩家获胜）。根据引理 XIII.19，对于任何赋值 β： $V_i \to \{0,1\}$，要么存在一个全称策略，能够在 $F[\beta]$中禁止 $\sigma \in \{e, \neg e\}$，要么存在一个存在策略，能够强制（从而允许）在 $F[\beta]$ 中实现 σ。也就是说，存在一个部分存在策略，允许在 $F[\beta]$中实现 e，并且存在一个部分存在策略，允许在 $F[\beta]$ 中实现 $\neg e$，这意味着 $\mathcal{T}$ 在 i处留下了一个选择。关于留下选择的部分存在策略与赋予能力的部分全称策略之间对偶性的证明是类似的。□

每个剩余分量都可以通过以下索引来刻画：存在全称策略在该索引处留下选择的最高索引 i，以及存在全称策略在该索引处赋予权限的最低索引 i。

定义 XIII.15。设 F 是一个布尔公式，且 var(F) $\subseteq$ var(Q)。如果存在一个部分全称策略，在某个索引 i 处留下选择，且 $0 \le i \le \ell$ 在 F 中，则令 $LC(F)$ 为满足条件的最大索引 i。类似地，如果存在一个部分全称策略，在索引 i 处赋予权限，且 $0 \le i \le \ell$，则令 $EE(F)$ 表示满足条件的最小索引 i。

这些索引使我们能够陈述剩余分量中存在全称获胜策略的充要条件。

引理 XIII.16。设 F 和 F' 是布尔公式，且 var(F) $\cap$ var(F') $\subseteq \{e\}$，var(F) $\cup$ var(F') $\subseteq$ var(Q)，满足

1）存在一个部分全称策略，在 F 中的某个索引 $0 \le i \le \ell$ 处留下一个选择；2）存在一个部分全称策略，在某个索引 $0 \le j \le \ell$ 处增强 F'。

于是，当且仅当 $EE(F') \le LC(F)$ 时，P.$F \wedge F'$ 存在获胜的全称策略。

证明。设 $EE(F') \leq LC(F)$。根据定义，存在一个部分全称策略，在 F 中的索引 $LC(F)$ 处保留一个选择。由此可知，存在一些部分全称策略，在每个索引 $0 \leq i < LC(F)$ 处都保留一个选择，尤其是在索引 $EE(F')$ 处。根据引理 XIII.13，P.$F \wedge F'$ 存在一个全称获胜策略。现在设 $LC(F) < EE(F')$。我们声称，在这种情况下，实际上必有 $LC(F)+1 < EE(F')$。为说明这一点，请注意，索引 $EE(F')$ 处的变量 v 必须出现在 F' 中。否则，在 F' 中索引 $EE(F')$ 处赋予能力的部分全称策略，就已经会在索引 $EE(F') - 1$ 处赋予能力，因为对 v 的赋值将无关紧要。但这样一来，$v \notin \mathrm{var}(F)$，而如果 v 是索引 $LC(F)$ 处的变量，我们就可以用关于变量 v 的任意函数扩展在索引 $LC(F)$ 处保留一个选择的部分全称策略，使其仍然在索引 $LC(F)+1$ 处保留一个选择，这就产生了矛盾。因此，该断言得证。根据引理 XIII.14，存在一个在 F 中索引 $LC(F)+1$ 处赋予能力的部分存在策略，以及一个在 F' 中索引 $EE(F') - 1$ 处保留一个选择的部分存在策略，这意味着必然存在一个在每个索引 $i \leq EE(F') - 1$ 处都保留一个选择的部分存在策略。根据上述已证明的断言，这包括索引 $LC(F)+1$，由引理 XIII.13 可知，P.$F \wedge F'$ 存在一个存在玩家的获胜策略。

□

只有一个分量 F_i 具有会留下一个选择的全称策略，此时可以使用蛮力法计算索引$LC(F_i)$。然而，不能简单地使用蛮力法计算索引$EE(\bigwedge_{j\neq i} F_j)$，因为该子公式中的分量数量不受常数 c 的限制。下一个引理表明，可以根据各个分量的索引计算该索引。

引理 XIII.17. 设 F 和 F' 为布尔公式，且 $\mathrm{var}(F) \cap \mathrm{var}(F') \subseteq \{e\}$，以及 $\mathrm{var}(F) \cup \mathrm{var}(F') \subseteq \mathrm{var}(Q)$，使得 $EE(F)$ 和 $EE(F')$ 有定义。则

$$EE(F \wedge F') = \min(EE(F), EE(F')).$$

证明。一个在 F 中的索引 i 处赋予权能的部分全称策略 $\mathcal{T}$，也会在 $F \wedge F'$ 中的索引 i 处赋予权能，因此 $EE(F \wedge F') \leq \min(EE(F), EE(F'))$。对于另一个不等式，假设 $\min(EE(F), EE(F')) > 0$，并令 $0 \leq i < \min(EE(F), EE(F'))$。根据引理 XIII.14，存在一个部分存在策略 $\mathcal{T}$，它在 F 中的索引 i 处留下一个选择，并且存在一个部分存在策略$\mathcal{T}'$，它在 F' 中的索引 i 处留下一个选择。相应地，根据引理 XIII.12 和引理 XIII.6，存在一个部分存在策略 $\mathcal{T}''$，它在$F \wedge F'$中的索引 i 处留下一个选择。再次应用引理 XIII.14，我们可以得出结论：不存在一个能够赋予权能的部分全称策略在 $F \wedge F'$ 中的索引 i 处赋予权能，因此 $\min(EE(F), EE(F')) \leq EE(F \wedge F')$。

□

由于我们所关注的策略相对于分量 F_i 的性质，都与不出现在 F_i 中的变量无关，因此可以通过对局限于分量的策略进行蛮力搜索枚举，来确定是否存在具有这些性质的策略。

引理 XIII.18. 设 F 是一个布尔公式，满足 $\mathrm{var}(F) \subseteq \mathrm{var}(Q)$。以下问题可在时间 $f(|\mathrm{var}(F)|)$ 内判定，其中 f是某个函数：

1) 是否存在一种部分全称策略（分别为存在策略），在 F 中禁止（允许） $\Sigma \subseteq \{e, \neg e\}$？ 2) 是否存在一种部分全称策略（存在策略），在索引 i 处通过 $0 \leq i \leq \ell$ 在 F 中赋予.....权能？ 3) 是否存在一种部分全称策略（存在策略），在索引 i 处通过 $0 \leq i \leq \ell$ 在 F 中保留一个选择?

定理 VI.5. 设 Q 是一个大小为 1 的 c-删除集合格范式量化布尔公式。判定 Q 是否为真的问题，以 c 为参数，是固定参数可处理的。

证明。令 e 表示删除集中的变量。不失一般性地，假设它经过存在量化，并且位于 Q 的前缀最内层。算法 A 首先判断是否存在一个具有全称获胜策略的分量；若存在，则输出 0。否则，它移除任意一个存在部分策略能够允许 e 和 $\neg e$ 的分量；根据引理 XIII.9，这不会改变真值。若剩余分量不超过一个，算法输出 1。否则，算法判断是否存在一些分量及相应的全称策略，使其分别在这些分量（不一定互不相同）中禁止 e 和 $\neg e$。若不存在这样的情形，算法输出 1；根据引理 XIII.4，这是正确的。若存在这样的分量且它们互不相同，则 A 输出 0；根据引理 XIII.8，这是正确的。令 F_i 表示这样的分量：其中存在全称策略，分别禁止 e 和 $\neg e$。对于剩余分量 F_j，算法计算 $LC(F_i)$ 和 $\min_{j\neq i}\{EE(F_j)\} = EE(\bigwedge_{j\neq i} F_j)$。根据引理 XIII.18 和引理 XIII.17，这可以在时间 $f(c)p(\|Q\|)$ 内完成，其中 f 和 p 是适当的函数。最后，若 $EE(F_i) \leq \min_{j\neq i}\{EE(F_j)\}$，则 A 输出 0，否则输出 1；根据引理 XIII.16，这是正确的。

□

C. 一种适用于每种类型含有多个分量的公式的算法

在此，我们为定理 VI.7 给出形式化证明。首先介绍以下概念。设 $Q = Q_1 v_1 Q_2 v_2 \cdots Q_n v_n . F$是一个合取范式量化布尔公式，且 $D \subseteq \mathrm{var}^{\exists}(Q)$ 是 Q的一个 c-删除集。考虑 D中变量的一个赋值β： $D \to \{0,1\}$。我们称存在策略 $\mathcal{T}$强制 β，如果对于每一个全称

对局 δ: $\mathrm{var}^{\forall}(Q) \to \{0,1\}$，都有 $\alpha(\mathcal{T},\delta)$ 扩展β。此外，如果对于每一个扩展 β的存在性博弈δ， $\alpha(\Lambda,\delta)$ 均不满足 F，则称全称策略 Λ 禁止 β。

定理 VI.7 的证明背后的主要思想如下。首先，我们证明：对于 $G_Q - D$ 的每个分量C 以及每个赋值 β: $D \to \{0,1\}$，其中赋值涉及删除集 D 中的（存在玩家的）变量，对于公式 $Q(D \cup C)$，以下结论成立：存在玩家要么拥有一种获胜策略，始终采用赋值 β；要么全称玩家拥有一种策略 Λ，能够阻止 β，也就是说，任何与 Λ 对局、并采用 β 的存在性博弈，都会输给 Λ。要判断Q 是否可满足，我们首先针对每种分量类型 t 计算集合 $A(t)$。这里， $A(t)$ 是所有赋值β： $D \to \{0,1\}$ 的集合；对于公式 $Q(D\cup C_t)$，存在玩家拥有一种能够迫使 β 的获胜策略，其中 C_t 是任意一个类型为 t 的分量。注意，集合$A(t)$可以高效计算，因为分量类型的数量以及公式 $Q(D \cup C_t)$ 中变量的数量，都受参数的某个函数所界定。随后我们证明， Q 可满足，当且仅当 $\bigcap_{t\in\mathcal{T}(Q,D)} A(t) \neq \emptyset$。这是因为，如果存在一个 $\beta \in \bigcap_{t\in\mathcal{T}(Q,D)} A(t)$，那么存在玩家就可以在每个分量中采用其迫使 β 的获胜策略，从而获胜。此外，如果不存在这样的 β，那么全称玩家就拥有每种分量类型的足够多副本，足以阻止存在玩家对删除集变量采取任何可能的赋值 β。

因此，以下引理对我们的算法至关重要。

引理 XIII.19. 存在一个迫使 β 的存在玩家获胜策略，当且仅当不存在禁止 β 的全称策略。

证明。为简化记号，我们令 $V = \mathrm{var}(Q)$。为证明正向，设 $\mathcal{T} = (\tau_v : \{0,1\}^{V^{\forall}_{<v}} \to \{0,1\})_{v\in V^{\exists}}$ 是一个迫使 β 的存在玩家获胜策略，并反设存在一个全称策略 $\Lambda = (\lambda_v : \{0,1\}^{V^{\exists}_{<v}} \to \{0,1\})_{v\in V^{\forall}}$，它禁止 β。令 δ: $V \to \{0,1\}$ 为 $\mathcal{T}$ 与 Λ 对局后得到的赋值，即 $\delta = (\alpha(\mathcal{T},\Lambda))$。于是， $\alpha(\Lambda,\delta^{\exists})$ 不满足 F，因为 $\delta^{\exists}$ 扩展了 β，且 Γ 禁止 β。然而，$\alpha(\Lambda, \delta^{\exists}) = \alpha(\mathcal{T},\delta^{\forall})$ 也满足 F，因为 $\mathcal{T}$ 是存在玩家的获胜策略，矛盾。

为证明反向，我们将构造一个存在玩家的获胜策略 $\mathcal{T} = (\tau_v : \{0,1\}^{V^{\forall}_{<v}} \to \{0,1\})_{v\in V^{\exists}}$，使其能够迫使 β，具体如下。我们沿着量词前缀从左到右迭代地构造 $\mathcal{T}$。也就是说，令 $v \in V^{\exists}$，并假设我们已经定义了部分存在策略 $\mathcal{T}_v = (\tau_u : \{0,1\}^{V^{\forall}_{<u}} \to \{0,1\})^{u\in V^{\exists}_{<v}}$，即 $\mathcal{T}_v$ 是针对所有出现在前缀中 v 之前的存在变量定义的部分存在策略。令 U 为关于 Q 的所有全称策略构成的集合。此外，对于赋值 δ: $V^{\forall}_{<v} \to \{0,1\}$，令 U_δ 为所有针对部分定义的策略 $\mathcal{T}_v$ 采取 δ 的全称策略构成的集合；也就是说，对于每个 $\Lambda \in U_\delta$，都有 $\alpha(\mathcal{T}_v,\Lambda) = \delta$。我们将保持以下不变式：

(*) 对于每个 δ： $V^{\forall}_{<v} \to \{0,1\}$ 以及每个 $\Lambda \in U_\delta$，都存在一个对局 $\delta_\exists$： $V^{\exists}_{\geq v} \to \{0,1\}$，它扩展了 β，使得 $\alpha(\Lambda, \delta' \cup \delta_\exists)$，其中 $\delta' = \alpha(\mathcal{T}_v,\Lambda)^{\exists}$ 满足 F。

由于我们假设不存在禁止 β 的全称策略，因此可得 U 满足 (*)。此外，如果不变式在末尾成立，那么 $\mathcal{T} = \mathcal{T}_n$ 就是存在玩家迫使 β的获胜策略。

现在，令 $v \in V^{\exists}$，并假设我们已经定义了 $\mathcal{T}_v$，即对于所有 $u \in V^{\exists}_{<v}$都有 τ_u。令 δ： $V^{\forall}_{<v} \to \{0,1\}$。注意，如果 $v \in D$，那么令 $\tau_v(\delta) = \beta(v)$，根据归纳假设即可得到 $\mathcal{T}_v$ 的一个满足 (*) 的扩展。因此，假设 $v \notin D$。注意，如果存在一个赋值 δ_v： $v \to \{0,1\}$，使得对于每个 $\Lambda \in U_\delta$，都存在一个对局 $\delta_\exists$： $V^{\exists}_{>v} \to \{0,1\}$，它扩展了 β，使得 $\alpha(\Lambda, \delta' \cup \delta_v \cup \delta_\exists)$，其中 $\delta' = (\alpha(\mathcal{T}_v,\Lambda))^{\exists}$确实满足 F，那么我们可以将 τ_v 对δ 的响应定义为 δ_v。接下来我们证明，这样的赋值 δ_v： $v \to \{0,1\}$ 必然总是存在，从而完成证明。

反设结论不成立，那么对于每个赋值 δ_v: $v \to \{0,1\}$，都存在某个$\Lambda^{\delta_v} \in U_\delta$，使得对于每个对局 $\delta_\exists$: $V^{\exists}_{>v} \to \{0,1\}$，只要它扩展了 β，赋值 $\alpha(\Lambda^{\delta_v}, \delta' \cup \delta_v \cup \delta_\exists)$（其中$\delta' = \alpha(\mathcal{T}_v,\Lambda^{\delta_v})^{\exists}$）就不满足 F。

现在考虑策略 Λ^{δ_v} 的如下合成 Λ，其定义如下：

- 对于每个 $u \in V^{\forall}_{<v}$，令 $\lambda_u = \lambda^{\tau}_u$，其中 τ 是将 v 赋值为 0 的赋值。
- 对于每个 $u \in V^{\forall}_{>v}$，令 $\lambda_u(\delta' \cup \delta_v \cup \delta'') = \lambda^{\delta_v}_u{}_{\delta' \cup \delta_v \cup \delta''}$ δ' $V^{\exists}_{<v} \to \{0,1\}$ ()，对于每个：，每个 δ_v: $v \to \{0,1\}$，以及每个 δ'': $V^{\exists}_{>v} \cap V^{\exists}_{<u} \to \{0,1\}$。

然而， $\Lambda \in U_\delta$ Λ 不满足 (*)，这与我们的归纳假设矛盾。 □

定理 VI.7。设 Q 是一个合取范式量化布尔公式，且 $D \subseteq \mathrm{var}^{\exists}(Q)$ 是 Q 的全称完备 c-删除集。那么，判定 Q 是否可满足是固定参数可处理的，其参数为 $|D| + c$。

证明。对于每个 $t \in \mathcal{T}(Q,D)$，令 $A(t)$ 为所有赋值 β: $D \to \{0,1\}$ 的集合，使得存在玩家对 $Q(D\cup C_t)$ 有获胜策略，能够迫使β 满足某个 $C_t \in \mathcal{C}(Q,D,t)$。我们声称， Q 可满足，当且仅当 $\bigcap_{t\in\mathcal{T}(Q)} A(t) \neq \emptyset$。请注意，这就完成了定理的证明，因为判断 $\bigcap_{t\in\mathcal{T}(Q)} A(t) \neq \emptyset$ 是否成立，可以通过以下方式在以 $|D| + c$ 为参数的固定参数可处理时间内完成。首先，我们使用命题 XIII.2 在多项式时间内计算 $\mathcal{T}(Q,D)$ 和 $\mathcal{C}(Q,D,t)$

对于每个 $t \in \mathcal{T}(Q, D)$。然后，对于至多 $\|Q\|$ 种类型 $t \in \mathcal{T}(Q, D)$ 中的每一种，以及至多 $2^{|D|}$ 个赋值 β: $D \to \{0,1\}$ 中的每一个，我们判断存在玩家是否对 $Q(D \cup C_t)$ 有获胜策略，能够迫使 β 满足某个 $C_t \in \mathcal{C}(Q, D, t)$；这至多需要 $\mathcal{O}(2^{|D|+c})$ 时间，可以通过穷举实现，因为 $Q(D \cup C_t)$ 至多有 $|D| + c$ 个变量。

为了证明正向，我们证明其逆否命题。也就是说，假设 $\bigcap_{t \in \mathcal{T}(Q)} A(t) = \emptyset$，我们证明其不可满足。根据引理 XIII.19，对于每个 $t \in \mathcal{T}(Q)$ 和每个 β: $D \to \{0,1\}$，若 $\beta \notin A(t)$，则存在一个全称策略 $\Gamma^{t,\beta}$，用于 $Q(D \cup C_t)$，该策略会禁止β，其中 $C_t \in \mathcal{C}(Q, D, t)$。由于 Q 是全称完备的，对于每个这样的 t 和 β，我们都可以选择一个类型为 t 的私有分量 $C^{t,\beta}$。现在令 Γ 为全称策略，它在分量 $C^{t,\beta}$中执行 $\Gamma^{t,\beta}$。于是，由于 $\bigcap_{t \in \mathcal{T}(Q)} A(t) = \emptyset$，我们得到：对于每个 β: $D \to \{0,1\}$，至少存在一个分量 $C^{t,\beta}$，使得 Γ 在 $Q(D \cup C^{t,\beta})$ 中执行 $\Gamma^{t,\beta}$。换言之，Γ 会禁止所有可能的 β: $D \to \{0,1\}$，因此Γ 是全称玩家在 Q 上的获胜策略，这表明 Q 不可满足。

为证明反向，设 $\beta \in \bigcap_{t \in \mathcal{T}(Q)} A(t)$。于是，对于每个分量 $C \in \mathcal{C}(Q, D, t)$，在 $Q(D \cup C)$ 中都存在一个存在玩家的获胜策略 $\mathcal{T}^{C,\beta}$，该策略迫使 β。不难验证，所有 $\mathcal{T}^{C,\beta}$'s 的合成是 Q 的存在玩家的获胜策略。 □